ESD-TR-88-314

Annual Report

Knowledge-Based System Analysis and Control Defense Switched Network Task Areas

30 September 1988

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS

Prepared for the Department of the Air Force under Contract F19628-85-C-0002.

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. The work was sponsored by the Department of the Air Force under Contract F19628-85-C-0002.

The ESD Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Hugh L. Southall

Hugh L. Southall, Lt. Col., USAF
Chief, ESD Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

KNOWLEDGE-BASED SYSTEM ANALYSIS AND CONTROL DEFENSE SWITCHED NETWORK TASK AREAS

ANNUAL REPORT SUBMITTED TO
DR. SYED SHAH
DCEC R610
1860 WIEHLE AVENUE
RESTON, VA 22090-5500

H.M. HEGGESTAD

Group 21

1 OCTOBER 1987 — 30 SEPTEMBER 1988

ISSUED 11 JULY 1989

LEXINGTON MASSACHUSETTS

ABSTRACT

An Interactive Defense Switched Network Simulator (IDSIM) has been implemented, consisting of an enhanced Call-by-Call Simulator (CCSIM) in one computer interfaced with a Network Management Expert System (NMES) in a second computer. The operation of IDSIM is similar to that of a real-world theater Defense Switched Network (DSN) and its community of users, at a future time when the DSN is fully installed and an Expert System at each theater operations center performs DSN network management (NM) functions. Within IDSIM, NMES collects periodic activity reports from each simulated DSN switch in CCSIM, analyzes them to identify network problems, and applies control commands to the simulated switches to circumvent the problems as well as possible. By inducing network fault and overload conditions, applying controls, and studying the results, an experimenter can exploit IDSIM as a system engineering tool to develop NM strategies for the DSN. This report describes IDSIM as well as a set of NM study results obtained with this tool. The report also describes the applicability of CCSIM as a near-term training device for human DSN NM operators, and analyzes expert system applicability to Defense Data Network (DDN) management.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION AND SUMMARY

This section makes reference to each of the task areas in Lincoln's FY88 Statement of Work from DCEC, and briefly summarizes the work accomplished in each area. The organization of this report does not precisely follow that of the Statement of Work (SOW), since the complexion of the research area has changed somewhat as the work has progressed. The following paragraphs key the SOW tasks to the report sections covering them.

Task I in the SOW requires the implementation of an Interactive Defense Switched Network Simulator (IDSIM), consisting of an enhanced Call-by-Call Simulator (CCSIM) in a SUN 3/260 workstation interfaced with an enhanced Network Management Expert System (NMES) in a second SUN 3/260 workstation whose primary function is to control the enhanced CCSIM. The design and implementation of IDSIM have been major parts of our FY88 efforts, and have resulted in a powerful system engineering tool intended to support rapid, efficient, hands-on investigation and development of Network Management (NM) strategies for the DSN. (NM here and throughout this report refers only to the real-time surveillance and network control functions that motivated the development of IDSIM. Other aspects of network management such as traffic engineering, accounting, and maintenance can be addressed by more traditional means, and are not supported by IDSIM.) Section 2 of this report describes the progress and status of all components of IDSIM.

Subtask IA of the SOW specified the development of an enhanced CCSIM. Enhancements carried out under this subtask are reported in Section 2.1. Significant CCSIM changes implemented this year include: a complete rework of the call data structure; basic reorganization of the way CCSIM communicates with other programs in IDSIM; modifications and additions of certain control commands to match NTI DMS-100 switch implementations; addition of a simulator command option to change offered traffic during a run (e.g., to introduce focused overloads or other anomalies); and augmentation of statistics matrix selections to report call intentions, failures, and mean tries to success for each source-destination pair, in contrast with the earlier output of these data as combined averages over all sources and destinations. Another significant FY88 work item was validation of CCSIM by calibration against Katz, the system of computer/analytic tools in use for some years by DCEC for AUTOVON and DSN engineering. Task IA also asked for the ability to simulate Common-Channel Signaling (CCS) as well as the current in-band signaling. Since this is basically a matter of restoring a CCSIM capability that was turned off at the start of the DSN NM investigations, and since CCS simulations are not yet required in our current activity, the extra work of restoring and testing CCS was deferred (by verbal agreement with DCEC) until needed in FY89. A consequence of this agreement was that the FY87 SOW wording about CCS was transferred verbatim into the FY89 SOW. Subtask IA also asked that CCSIM preserve the capability originally designed into it, to simulate satellite Demand Assigned Multiple Access (DAMA); this has been done, and when a DAMA implementation for the DSN is defined in the future it can be incorporated into CCSIM.

Subtask IB of the SOW specified the development of an enhanced Network Management Expert System (NMES). Work in this area has progressed as specified, and is reported in Section 2.3. Related

knowledge engineering work is reported in Section 3. The subtask specified expansion of the NMES knowledge base, development of an interactive interface with CCSIM, and development of operating features providing for convenient use of IDSIM for network management investigations. These goals have been realized in the course of the year, as described in the following paragraphs. During FY88, the NMES was developed from the FY87 prototype expert system into a fully integrated component of IDSIM capable of receiving, processing, and analyzing switch report information from CCSIM and sending network management controls back to it. NMES has progressed to include many additional monitors, a problem detection module, a confirmation module, and a planning module. The additions to the functional architecture have centered around the detection of damaged nodes, restored nodes, overloaded trunk groups, trunk groups that are no longer overloaded, damaged trunks, and restored trunks. We expect that these NMES capabilities, as a minimum, will be demonstrated at DCA-Europe in 4Q FY89.

Subtask IC of the SOW specified the development of a graphics interface for IDSIM to support DSN system engineering and also to demonstrate the feasibility of using IDSIM as a DCOSS controller trainer. Section 2.2 describes the powerful new graphics interface that has been developed for IDSIM in FY88 in response to the first requirement of this subtask. CCSIM experiments can now be controlled either from the graphics interface or from a prestored file of commands. The graphics interface is preferred for demonstrations and for experimental use in situations where the experimenter is seeking insight into a network problem by watching a scenario develop. As work progressed on the DCOSS controller trainer feasibility demonstration specified in Task III, it became clear that the use of IDSIM to emulate DCOSS graphics was not the preferred approach to achieving a trainer capability. When an alternate approach was chosen for the trainer, we abandoned any work relative to the second requirement of Subtask IC. An integrated discussion of the whole DCOSS controller trainer topic is given in Section 4.

Task II in the FY88 Statement of Work provided for performing traffic overload/congestion/damage simulation studies with IDSIM. One purpose of these studies is to develop knowledge of switch statistics report patterns corresponding to classes of such network problems, leading to the ability to recognize incipient problems before conditions become critical. The other main purpose of the simulation studies is to learn how to respond effectively to these problem categories by selecting and applying NM control actions. Both kinds of knowledge are steadily being added to the knowledge base of NMES through implementation of new monitors and rules, as described in Section 2.3 of this report. FY88 accomplishments in the simulation studies area include: considerable further scrutiny of the ramifications of switch damage scenarios and appropriate control actions; evaluation of the use of the CANcel-To (CANT) control as a response to switch outages; experiments with routing changes to improve call failure rate under certain switch outage scenarios; studies of the damaged trunk group and noisy trunk group problems; studies of two kinds of focused overload conditions (high overall traffic to/from a switch or region, and high traffic to a few busy destinations); and studies of detection and correction of nonfocused trunk group overload conditions. Our observations resulting from the simulation studies are presented in Section 3. While it is attractive to imagine encapsulating the results of our simulation studies in a crisp set of prescriptions coupling clearly recognizable patterns with specific problem types and precise

recipes for control actions to correct them, we conclude that the situations in which simple problem-response prescriptions are valid are in the minority. Problem recognition is often more straightforward than is the formulation of a plan for NM actions to deal with the problem.

Task III of the FY88 SOW required demonstration of the feasibility of creating a DCOSS controller training device. This work was completed early in the year, and DCA asked Lincoln Laboratory to begin actually implementing such a device. Section 4 combines the reporting of the work under Task III with a description of the status of the DCOSS Controller Trainer development. The latter activity has shifted from a focus on the DCOSS design developed by the Air Force Sacramento Air Logistics Center, to an interim capability based on the Network Management Support System (NMSS) developed by GTE for DCA-Europe. Our work involves extensions to CCSIM plus the development of a Trainer interface to NMSS. The expected capabilities of the Trainer and its interfacing to NMSS are described in Section 4.

During the course of the year, DCA asked for a field test and evaluation of NMES to be planned for late FY89 at DCA-Europe using the same input data available to controllers using NMSS. This new requirement has considerably influenced our FY88 activity, in terms of both making NMES implementation choices and planning our FY89 work toward the demonstration objectives. Two Lincoln staff spent a week at DCA-Europe in September coordinating needs and plans with the site personnel, for both the NMES demo and the DCOSS Controller Trainer, which is a closely related activity because we will be using the Trainer to provide inputs for NMES during the development of the field-test version. Section 5 describes our design for the field-test system.

Task IV of the FY88 SOW provides for analysis and review of future requirements of the Defense Data Network (DDN) for monitoring and control, and for identification of expert system techniques and technology that could be applied to improve the effectiveness of DDN management. This work is reported in Section 6. It was carried out primarily in the first half of the year, and an interim report was presented at a briefing at DCEC on 10 March 1988. Section 6 describes the work and summarizes the recommendations we developed. Basically, of the seven kinds of activity at DDN Monitoring Centers, we found two (resource management and crisis management) requiring complex situation analysis and choice of corrective action plans in real time, such that expert system technology could be applied (at considerable cost) to enhance operator performance. In both cases, augmentation of the expert system with an operator training simulator would be very useful. As explained in Section 6, the other five kinds of activity could benefit considerably by adding more conventional automation and database management systems, but we found no clear justification for expert system application.

Appendix A of this report summarizes the definitions, syntax, and application of all the NM controls currently implemented in IDSIM. Near-term NM control recommendations are listed in Appendix B. The user's guide documents that have been issued are described in Appendix C.

2. THE INTERACTIVE DSN SIMULATOR (IDSIM)

Task I in the Statement of Work requires the implementation of an Interactive Defense Switched Network Simulator (IDSIM), consisting of an enhanced Call-by-Call Simulator (CCSIM) in a SUN 3/260 workstation interfaced with an enhanced Network Management Expert System (NMES) in a second SUN 3/260 workstation whose primary function is to control the enhanced CCSIM. The design and implementation of IDSIM have been major parts of our FY88 efforts, and have resulted in a powerful system engineering tool intended to support rapid, efficient, hands-on investigation and development of Network Management (NM) strategies for the DSN. (NM here and throughout this report refers only to the real-time surveillance and network control functions that motivated the development of IDSIM. Other aspects of network management such as traffic engineering, accounting, and maintenance can be addressed by more traditional means, and are not supported by IDSIM.)

The development of the enhanced CCSIM was carried out as specified in Subtask IA of the SOW, and is reported in Section 2.1 below. A powerful and flexible graphics interface has been developed as an alternative to command files for controlling CCSIM experiments, and Section 2.2 has been devoted to describing its features. The development of the enhanced NMES (Subtask IB in the SOW) has progressed as specified, and is reported in Section 2.3. Subtask IC of the SOW originally envisioned use of IDSIM graphics displays to represent those of DCOSS, as part of the DCOSS operator trainer feasibility demonstration specified in Task III. As the work progressed, the design for the trainer came into sharp focus, and its actual implementation was begun; an integrated discussion of the whole topic (including the resolution of Subtask IC) is given in Section 4.

IDSIM is made up of four cooperating computer programs which operate as individual communicating processes running in one or more SUN 3/260 workstations. The four processes are:

(1) A Call-by-Call Simulator (CCSIM) that simulates the network being investigated as well as the behavior of its users.

(2) A graphics interface that allows the IDSIM user to control the simulation and observe the behavior of the network and its performance as seen by its simulated users. NM controls can also be applied and removed through the interface.

(3) A command file interpreter that provides an alternative means of controlling a simulation from a preplanned file of commands. This capability allows for unattended operation of CCSIM as well as providing a convenient mechanism for introducing complex sequences of commands that might be needed repeatedly.

(4) A Network Management Expert System (NMES) that watches the course of the simulation and can automatically apply and remove network controls when it recognizes that situations calling for such actions have occurred. The IDSIM user can inhibit the action of NMES when desired, and can adjust the parameters used in the expert system rules for recognizing problems and specifying control responses.

The four processes communicate using TCP/IP packet protocols over an ethernet local-area network that interconnects the workstations. For experiments that do not involve NMES, IDSIM performs well with the remaining three processes executing on a single workstation. However, since NMES is a large program that makes relatively heavy demands for CPU cycles, performance is more satisfactory if NMES is run on a separate workstation. The use of the second workstation gives the further advantage of allowing both the simulator graphics and the expert system graphics to be visible concurrently.

The design of IDSIM does not limit the number of intercommunicating processes. As will be discussed in Sections 4 and 5, in FY89 we are planning to add a fifth process which will allow IDSIM to function as a training aid for DCOSS controllers and to support a field demonstration of NMES.

Delivery of IDSIM to DCEC has taken place in stages during FY88, in order to make these software tools available to DCEC personnel as rapidly as possible. This report applies to the software versions in place as of October 1988, although results obtained by use of IDSIM in the course of the year remain valid, since successive releases of IDSIM have corrected minor problems and refined some features, rather than changing basic functionality. The first delivery was that of two SUN 3/260 workstations in January 1988. Lincoln personnel installed the operating system and the then-current version of CCSIM on these machines, as well as porting to the SUNs the 1985 version of CCSIM that DCEC personnel had been using on another computer since the end of the DCEC-sponsored Experimental Integrated Switched Network (EISN) program. In March 1988, a new CCSIM release was sent to DCEC on tape and was successfully installed by DCEC personnel. Preliminary versions of NMES and the graphics interface were demonstrated at DCEC in May, June, and July, and delivery versions were installed in August and released for use by DCEC personnel. This report refers to the new release of IDSIM undergoing testing and preparation at Lincoln Laboratory in October, and scheduled for delivery to DCEC in November 1988.

Although IDSIM has been in a constant state of development and change throughout the year and will continue to be so in the next year, it has been sufficiently stable for DCEC personnel to do useful work with the system. DCEC personnel required only a few days of very informal training by Lincoln staff during the course of installation and project review site visits.

2.1. THE DSN CALL-BY-CALL SIMULATOR (CCSIM)

Important CCSIM changes implemented this year under Task IA include: a complete rework of the call data structure; basic reorganization of the way CCSIM communicates with other programs in IDSIM; modifications and additions of certain control commands to match NTI DMS-100 switch implementations; addition of a simulator command option to change offered traffic during a run (e.g., to introduce focused overloads or other anomalies); and augmentation of statistics matrix selections to report call intentions, failures, and mean tries to success for each source-destination pair, in contrast with the earlier output of these data as combined averages over all sources and destinations. Another important task item was validation of CCSIM by calibration against Katz, the system of computer/analytic tools in use for some years by DCEC for AUTOVON and DSN engineering. Task IA also asked for the ability to simulate Common-Channel Signaling (CCS) as well as the current in-band signaling. Since

this is basically a matter of restoring a CCSIM capability that was turned off at the start of the DSN NM investigations, and since CCS simulations are not yet required in our current activity, the extra work of restoring and testing CCS was deferred (by verbal agreement with DCEC) until needed in FY89. A consequence of this agreement was that the FY87 SOW wording about CCS was transferred verbatim into the FY89 SOW. Subtask IA also asked that CCSIM preserve the capability originally designed into it, to simulate satellite DAMA; this has been done, and when a DAMA implementation for the DSN is defined in the future it can be incorporated into CCSIM. All these changes are discussed in the following paragraphs.

CCSIM was originally developed at Lincoln Laboratory under the DCEC-sponsored EISN program to support a study of routing and preemption alternatives for the DSN, and was delivered to DCEC at the termination of work on EISN in September 1985. The original CCSIM was written in RATFOR (Rational Fortran) and consisted of about 27,000 lines of code. At the start of the current project in FY87, we began a major effort to extend and modify CCSIM to support the investigation of network management strategies for the evolving DSN. In the course of that year, we removed large sections of code that supported DAMA satellites and routing options such as precedence flooding that are not of current interest for the DSN. We extended CCSIM to model switch processing and signaling times in more detail and to account for trunk resources tied up by calls that block at points in the network beyond the source switch. We added a number of network management controls and periodic switch reports using the DSN Generic Switch Specifications supplied by DCEC as a guide. We also provided for engineered routing as used in the Pacific DSN and multiple trunk groups between pairs of switches. In-band signaling was introduced both in its impact on signaling delay and its effect on calls routed to a damaged switch.

CCSIM uses an event-by-event simulation technique to work out the fate of individual calls generated randomly with a Poisson arrival-time distribution and exponentially distributed holding times so as to achieve an average offered traffic pattern matching a given point-to-point erlang traffic matrix. The calls so generated are taken as a set of call intentions and are allowed to retry after blocking, preemption, damage, or reaching a busy destination according to parameters set by the experimenter. Statistics produced during the simulation allow the experimenter to observe both the performance of the network as seen by the switches [Grade-of-Service (GOS) as well as detailed trunk statistics] and also the performance as seen by its users (number of attempts needed to complete an intended call, or failure to do so). The latter information is not available for the real DSN or other telephone systems.

CCSIM has continued to evolve during FY88 and will change further during the upcoming year. The work has two goals. One is to provide a network research tool for use at DCEC and at Lincoln Laboratory in exploring routing and network management strategies. The other is to support network controller training and the field demonstration of the network management expert system (see Sections 4 and 5 of this report). Both goals require that CCSIM be able to faithfully simulate the DSN as it exists and evolves.

In the following subsections we summarize recent work on CCSIM and related activities. We include more detailed descriptions of some CCSIM features in Appendix A. The capabilities we describe are those of the October 1988 version, which we expect to deliver to DCEC in November 1988.

2.1.1. Changes in CCSIM in FY88

Much work was done during this year on the internal structure of CCSIM. The call data structure was completely redone. A call entering the system is now given a number and an associated place in the data structure that it retains through all retry attempts until it leaves the simulation. All facts about the call are recorded in arrays indexed by its call number. The change has greatly simplified the code, making it easier to keep the call statistics correct as new features are added, and it has saved approximately 200,000 bytes of memory and increased the speed of the simulation by a small amount.

By using the new call data structure, the retry statistics have been redone to give a better picture of network performance over specified time intervals. Although the retry behavior of all calls is watched continuously, the statistics are now reported only for those calls that leave the system during the interval of interest. These are calls whose fate is known. They have either completed successfully, were canceled by control action, or failed by exhausting their retry possibilities. Calls that are still in progress or are waiting to retry do not show in the printed summaries. The new output also contains an overall computation of the performance measures, Call Failure Rate (CFR) and Mean Tries for Success (MTFS) that we use for comparing network management control actions.

The method by which CCSIM communicates with other programs in IDSIM has been modified. These modifications have standardized the interface to CCSIM, reduced the overhead associated with sending messages to and receiving messages from the other programs, and provided an easily extendable protocol which isolates changes in one program from the others.

CCSIM now processes all of its input in a consistent manner and is no longer restricted to processing all the input from one source before accepting input from others. Should it become necessary in the future to have more programs communicate with CCSIM, adding them will be a relatively trivial task.

The messages consist of a header and a data portion. The header has a fixed format. The format of the variable-size data portion depends upon the message type which is contained in the header. Currently, 70 message types are defined, and the number is expected to grow substantially in the near future.

When starting up IDSIM, both the graphics interface and NMES are activated before the simulator. During its initialization, CCSIM attempts to establish connections with all the member programs of IDSIM. It writes logging messages indicating all successful connections. When the other components receive notice of successful connection to CCSIM, they begin operation — e.g., the graphics interface brings up its display and allows the user to enter commands. The simulation clock is allowed to advance only when all connected programs so indicate. Thus, either the human experimenter or the expert system can stop a simulation at will. In normal operation, NMES stops the simulation every 5 min of simulated time to allow it to digest the most recent complete set of switch reports. The user can stop CCSIM at will via the graphics interface, allowing him to observe the situation and to compose and enter commands at leisure.

The NM controls in CCSIM have been reworked from the generic switch specification forms originally generated, so that they now conform as closely as possible to those in the real DSN switches.

Appendix A has a complete list of the NM controls in the November delivery version of CCSIM. Based on study of Northern Telecom documentation as corroborated by DCEC, we now believe that these controls operate consistently with the DMS switches in the 1988-89 Pacific DSN, with the exception that trunk group controls in CCSIM generally operate on composite trunk groups combining all the satellite or terrestrial groups between pairs of switches. This combination makes control application simpler for an experimenter who tends to want the action to apply to a path rather than a particular trunk group. For the controller trainer and NMES demonstrations discussed in Sections 4 and 5, we will be changing CCSIM during the next year to apply the controls only to individual trunk groups because real switches expect to receive control commands in that form.

Because AUTOVON switches have a directionalization control called DRZ and because DCEC is interested in simulations involving these switches, we have included DRZ in CCSIM. CCSIM also implements a control called DRE that is used for similar purposes in DMS switches, but has a very different detailed behavior than DRZ. The differences are described in Appendix A.

During the year, we added the control CANF (CANcel-From) which cancels traffic overflowing from a trunk group. This is a control available in the current DSN. We also completed the implementation of two versions of ARC (Alternate Route Cancellation). These controls cancel alternate routes for calls and are included in the generic switch specifications, but are not currently available in DSN.

A useful feature added to CCSIM this year is a command to change the traffic offered to the network. It operates by multiplying elements of the traffic matrix by a factor stated in percent. The range is −100 to +99999 percent. The command is called LOAD-LEVEL and may be applied from an individual source to a single destination, from a source to all destinations, and from all sources to a single destination. Individual precedences may be specified, or the traffic changes may be applied equally to all precedences. By combining commands, quite arbitrary traffic patterns can be created with the exception that, if a source-destination pair had zero traffic in the original matrix, the command cannot cause traffic to appear.

Several new output matrices have been added to CCSIM to provide information about the distribution of call intentions, call failures, and the number of tries needed for successful calls for all source-destination pairs as well as combined values from all sources and to all destinations. The matrix printout capability was also extended to handle automatically the problem posed by large nets, for which the data will not fit on a single page.

2.1.2. Simulator Validation

Since CCSIM is an important tool for research in NM techniques both at Lincoln Laboratory and at DCEC, it is critical that its users have confidence in its validity in simulating the networks they are studying. Unfortunately, there is little real network data against which CCSIM outputs can be compared. Particularly in the critical area of detailed behavior of the switches when handling preemption, implementing controls, dealing with signaling failures on trunks, etc., we must depend upon study of switch documentation and discussion with experts, coupled with analysis of our programs, to convince ourselves

and other users of CCSIM that it is performing correctly. This is a tedious and difficult process which we started at the beginning of our work on the project, and we expect to continue until the end. We now have a full set of relevant Northern Telecom Practices that are applicable to DMS switches in the DSN. We have just begun the large task of digesting this mass of documentation and have already found some useful information on trunk signaling failures. We have no detailed information on the Siemens switches widely deployed in the European DSN, and will be working with DCEC to obtain such information when it becomes available so that we can complete the accurate modeling of the European DSN.

On a more encouraging note, we can report that we have conducted a series of calibration runs against some data provided by DCEC from Katz, the system of computer/analytic tools in use for some years by DCEC for AUTOVON and DSN network engineering. Katz has been validated by many comparisons between predicted network performance and actual data measured in the field, and it is reasonable to require that CCSIM should produce the same outputs as Katz, given identical inputs. This technical compatibility has now been established to the satisfaction of DCEC and Lincoln, as described below.

The problem of calibrating CCSIM with Katz is complicated by a number of factors which are not questions of correctness of implementation, but are inherent in the nature of the two different approaches. The most obvious issue is that the equations underlying Katz represent stochastic averages: their smoothness and precision correspond to average behavior over very large numbers of sample functions, or equivalently to time averages taken over extremely long runs under steady-state conditions. CCSIM outputs, on the other hand, are based on counts of discrete events triggered by random-number generators over runs that have to be finite in duration because of practical limitations on computer running time and the investigator's patience. The statistical parameters of the random processes in CCSIM are in fact chosen so that CCSIM's outputs would equal those of Katz, given extremely long averaging times; in practice, however, the results of any given finite-duration run have predictable random fluctuations.

To quantify this statement, consider the relationship between the specified traffic in erlangs from Node A to Node B, and the actual traffic computed from the measured statistics of a particular simulation run. In setting up the run, we divide this specified traffic figure by the average call duration to compute the average rate of call initiation from A to B; this number is the input to a random number generator in CCSIM which produces a Poisson sequence of call events. If N is the actual count of calls from A to B in time T for a particular simulation run, then we estimate the average call rate as N/T. It is easily shown that the normalized standard deviation of this estimate is equal to $1/\sqrt{N}$; that is, a call count of 100 may easily be in error by 10 percent, and a count of 10,000 may still have a 1-percent error. For typical CCSIM runs of a few hours of simulated time, it is not unusual to find call counts of a few dozen or less between some node pairs, and, therefore, CCSIM's estimates of offered traffic and GOS on lightly loaded links will be noisy compared with Katz's predictions. For heavily loaded links, on the other hand, CCSIM and Katz can be expected to be quite close. Indeed, both these effects are clearly discernible in the data examples below.

Another source of potential variation between Katz and CCSIM is the quantization that must be done in order to make simulation feasible. Time is divided in CCSIM into finite-duration intervals called clock ticks, and for each tick an event list is created by running random-number generators to determine

the times of occurrence of events. If the tick interval is too short, the number of computations becomes very large, so that the computer bogs down and runs slowly for even simple simulations. If the tick interval is too long, quantization noise becomes excessive. While our experience has indicated that a tick duration of 0.1 s is a generally satisfactory compromise, some quantization noise effects are inescapable, particularly when many events almost coincide in time. For example, two events are in conflict in CCSIM if they occur within the same clock tick interval, even though they might not have conflicted in the real world.

Because of all these factors, a significant amount of work was expended in identifying, understanding, and in some cases reducing discrepancies between Katz and CCSIM results. For example, the CCSIM call-generation routines had to be modified to round off start times rather than truncating them, and the ordering of event lists had to be altered to depend upon floating-point times, rather than integer clock tick counts. The outcome was successful completion of two sets of simulation runs using a 23-switch, 1989 version of DSN-Pacific. One set used a West-Pac busy-hour traffic matrix, and the other an East-Pac busy-hour matrix, for which Katz and CCSIM results matched satisfactorily. The CCSIM figures were obtained by averaging ten 10-h simulation runs using ten different random-number seeds, and the Katz figures were produced by the normal procedures at DCEC. Table 2-1 gives a representative example of the results, for East-Pac busy-hour traffic; it compares Katz and CCSIM predictions for all incoming and outgoing traffic for each switch in Hawaii and for the CONUS gateway switches at Lodi and San Luis Obispo, California. Notice that the GOS comparisons become noisy for values below about 1 percent and are closer for larger values, as expected in view of the explanations above. (Note also that the actual value of GOS is basically irrelevant if it is known to be very small.) The offered traffic figures are very close, as expected in view of the fact that the CCSIM figures are derived from large-event counts. For smaller DSN-Pacific nodes (not included in the table) with only a few erlangs of offered traffic, the percentage differences between Katz figures and CCSIM results are somewhat larger.

TABLE 2-1

Katz/CCSIM Calibration Examples

Node	GOS				Traffic (erlangs)			
	From		To		From		To	
	Katz	CCSIM	Katz	CCSIM	Katz	CCSIM	Katz	CCSIM
PRL	0.0131	0.0145	0.0025	0.0039	357.81	357.92	382.67	383.28
SCS	0.0085	0.0117	0.0029	0.0082	305.86	306.60	404.91	404.85
FSR	0.0237	0.0295	0.0039	0.0089	272.63	271.85	189.48	189.72
HIK	0.0229	0.0273	0.0013	0.0037	168.45	168.49	143.26	143.63
CSH	0.0247	0.0286	0.0019	0.0041	210.91	211.92	160.40	159.83
LOD	0.0149	0.0058	—	—	103.43	103.17	0.00	0.00
SLO	0.0019	0.0052	—	—	103.59	103.85	0.00	0.00

The detailed results of the simulation runs have been reviewed by DCEC R700 and accepted as showing agreement within expected error ranges.

It should be noted that the comparisons above represent a considerable amount of work and constitute only a zeroth-order validation of CCSIM. In order to be comparable with the Katz results, it was necessary to turn off preemption and retries in CCSIM, and to set equal to zero the holding time for trunk resources occupied by calls that block inside the network. As a result, we are able thus far to claim only that our routing matches that used by Katz, and that our call generators produce a good approximation to the desired traffic patterns.

Clearly, the more sophisticated features of CCSIM such as preemption and user behavior models are critical to network management studies for the DSN, and it is important for these features to be validated as well. Three mechanisms are contemplated for achieving this validation over time: (1) exposure of CCSIM to experimentation and scrutiny by knowledgeable DCA network engineers and operations personnel; (2) comparison of CCSIM with the real DSN as the latter evolves toward full implementation; and potentially (3) comparison of CCSIM with Northern Telecom's test bed of multiple real DMS-100 switches at Richardson, Texas. Mechanism (1) is being addressed through the availability at DCEC of the current version of CCSIM running on SUN workstations, where it is used for a variety of purposes. Both (1) and (2) will be greatly facilitated by the installation at DCA-Europe of the DCOSS Operator Trainer system which incorporates CCSIM. Mechanism (3) has great appeal, and DCEC and Lincoln Laboratory expect that it will be exploited. It should be possible to do some thinking and planning in this regard during FY89, but it is clear that the manpower resources to actually use the test bed will not be available until after the 4Q FY89 field demonstrations at DCA-Europe. Use of the NTI test bed will require careful study and extensive preparation; actual contact time is very tightly scheduled, and is billed at $10,000 per hour.

2.1.3. Old CCSIM Support

At the request of DCEC personnel, we moved the old (1985) version of CCSIM to the SUN workstations when we delivered them in January 1988. They were interested in running that version since it had some features of interest for their work that are not available in the current version of CCSIM. During the year, it developed that two features of the old version hampered its usefulness. One was that it applied an automatic code block to all calls to a damaged node. Another was that the reported link GOS statistics were incorrect. Using the ARPANET connections between Lincoln and DCEC, we were able to remove the unwanted code block and fix the statistics.

2.2. CCSIM CONTROL

We devote this section to describing the powerful new graphics interface developed for CCSIM in FY88. CCSIM experiments can now be controlled either from the graphics interface or from a prestored file of commands. The graphics interface is preferred for demonstrations and for experimental use in situations where the experimenter is seeking insight into a network problem by watching a scenario

develop. Control from a command file is useful for unattended runs where statistics gathering is the primary goal. The two modes can also be usefully combined by making a file with commands that are common to a series of runs and introducing the free variables from the graphics console.

We are working toward a capability that will allow the graphics interface to operate independently of CCSIM for viewing saved pictures of the network and the corresponding simulator statistics. This mode of operation should help in the analysis of experimental results which is currently slowed by the need to peruse rather large quantities of hard-copy outputs from CCSIM.

In the following subsections we describe the graphics interface and its features in some detail, and report briefly on the new command file processor that we have added to the IDSIM design.

2.2.1. Graphics Interface

The graphics interface is a C program which makes use of the SunView graphics and windowing software provided with the SUN workstation. A keyboard and mouse are used for input, and it is assumed that a color monitor is available for output. The style of interaction favors the mouse, which on the SUN has three buttons: left, right, and middle. Almost all user inputs can be introduced by making selections from menus using the mouse. The keyboard may be used as an alternative for entering node names, control values, etc. if the user prefers. Its use is required in only a few instances, such as adding new nodes to the map.

As shown in Figure 2-1, the output is divided into three areas. The large area at the top normally displays a map of the network with the links colored using data from switch reports according to options selected by the user. These options are described below. Alternatively, the large area can be used to display numerical statistics from CCSIM. The choices for these displays are also described below. The small area at the lower right is used for input via keyboard and mouse, and the small area at lower left shows a log of actions taken from the interface as well as any other components of IDSIM, e.g, NMES. The log window is scrollable so that the user can look back at the history of the simulation.

The primary control of a simulation experiment is exercised by placing the mouse anywhere in the map area and holding down the right mouse button. This action brings up a "pull right" type of menu that shows the action options available. For each action option, a list of possible sub-options is brought up by moving the mouse to the right. Moving the mouse vertically on the menu highlights the option that will be selected if the right mouse button is lifted in that position. The action options are:

> RUN SIMULATOR — Sub-options allow running for selected time periods, pausing, and resuming after a pause.
>
> MATRIX OPERATIONS — Sub-options allow displaying and clearing statistics matrices as well as sending them to the log file. The choice of which matrix to display is made using menu choices brought up in the lower right window. (See below.)
>
> SWITCH CONTROL — Sub-options select the NM control to be applied (see Appendix A for list of controls). Parameters for the selected control are entered using a control-specific menu brought up in the lower right window.

SIMULATOR CONTROL — Sub-options allow switch reports to be turned on and off and the traffic level in the simulation to be changed using a menu brought up in the lower right window to enter the parameters.

SUNTOOL — This option allows access to the underlying SUN window system. It is present as a convenience for the programmer, and is not needed for NM experiments.

There are eight choices for the colorings applied to the links in the network map. All have the common property of using an approximately continuous color scale in presenting switch report data. For data that can range from "good" to "bad" values, we use a green-through-yellow-to-red scale. For data such as trunk overflows that do not have values thought of as "good," we use a yellow-to-red scale. The scale used for each coloring is shown at the lower left of the network map together with the numerical values associated with the end points. When no data are present, a light blue color is used that does not belong to either continuous scale. We chose to use a continuous scale in the belief that we might be able to see patterns more easily than if there were abrupt color changes caused by small changes in the underlying numerical data. For example, if only a few colors are used, small differences near the boundaries will make large differences in the coloring, whereas larger differences away from the boundaries will have no effect on the colors. Such differences are useful when associated with alarm thresholds and the like, but our plan for IDSIM puts such thresholding action in NMES. We therefore have kept the simulator graphics as linear as possible.

For all the link colorings, we combine the data for the trunk groups that make up the link. Currently, there are at most two such groups — a terrestrial group and a satellite group. For the controller trainer and NMES demonstration applications of IDSIM discussed in Sections 4 and 5, we will have an arbitrary number of groups. We plan to extend the graphics interface to allow the data for the individual groups to be displayed when desired, but we expect that the overall map will continue to be colored according to combined data from all groups.

For the coloring, the link between any pair of switches is split into two segments, and the segment adjacent to each switch is colored according to data reported by that switch. Since the switch reports are not synchronized in time, it is not unusual for the two segments to have different colors displayed at the same time. It is possible to make a simple program change in CCSIM to cause the reports to all correspond to the same time period, but we do believe such synchronism is likely to be seen in a real network and have tried to avoid any dependency upon it in our work

The choice of coloring to be displayed is made my moving the mouse to one of the buttons at the top of the map display and pressing the left mouse button. The eight colorings are:

Usage — Usage is computed in the switch reports by sampling trunk occupancy every 100 s and accumulating the total for the three samples represented in each 5-min report. The occupancy values in CCSIM's switch reports are broken down by precedences. For the display, the values are combined to show usage for all precedences. For the coloring, we divide the reported usage by three times the number of working trunks in the group.

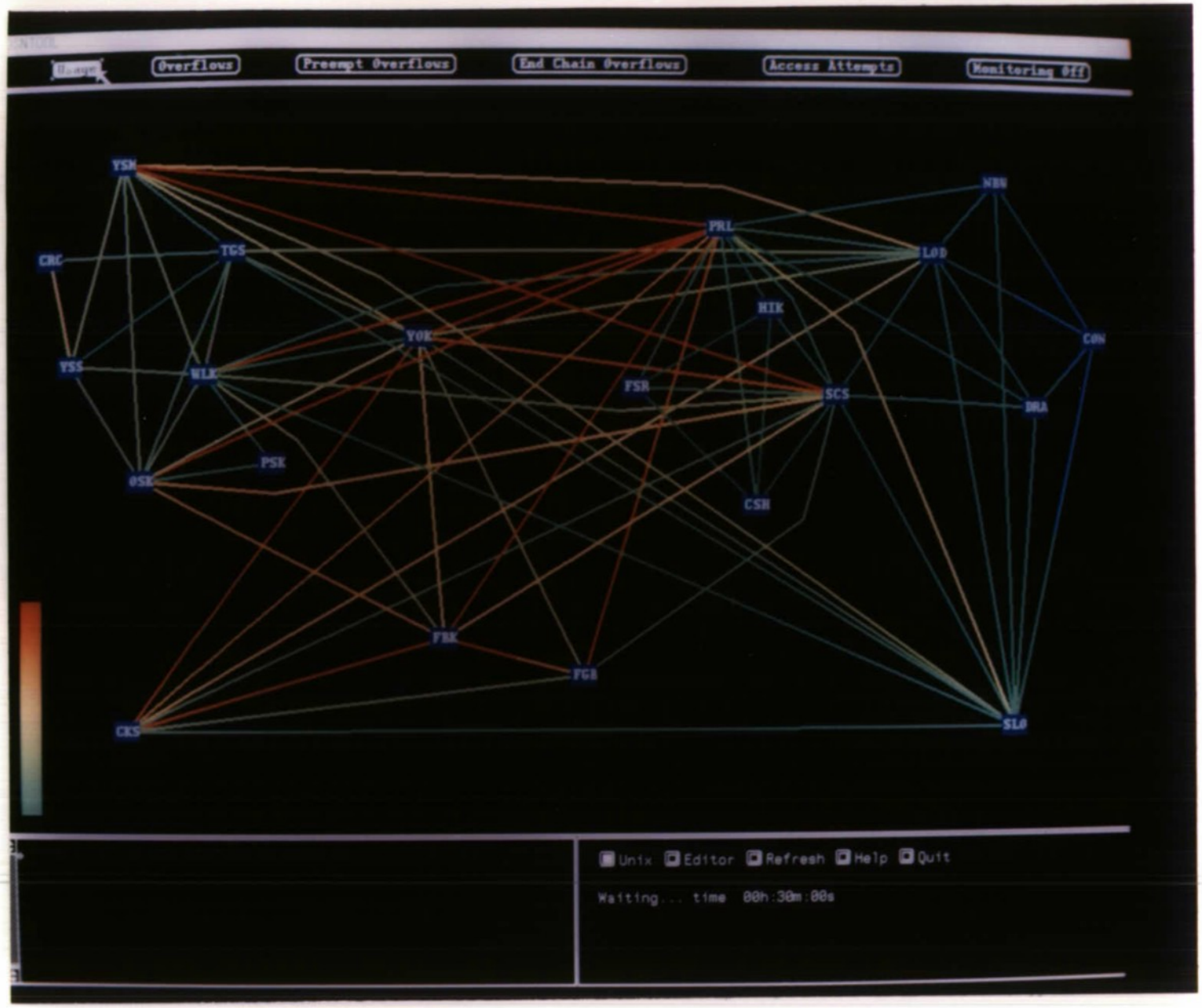


Figure 2-1. Photograph of IDSIM graphics interface display showing the 1988-89 Pacific DSN network. The simulation used West-Pac busy-hour traffic. Link coloring corresponds to trunk usage.

The resulting normalized usage is displayed on a green-to-red scale corresponding to 0- to 100-percent usage. When trunk losses occur at an arbitrary time within the switch reporting interval, it is possible that the reported usage may exceed 100 percent after division by a reduced number of working trunks. In such a case, the program truncates the value to 100 percent for the display.

Precedence Usage — This coloring is similar to the *Usage* coloring described above, except that it combines only the occupancy values for precedences higher than routine. This measurement is not currently available from the real DSN switches, but we have included it in IDSIM to see if it would be useful to have these data available.

Overflows — This coloring shows the ratio of reported trunk group overflows to attempts for all precedence levels combined. Values greater than zero are colored on a yellow-to-red (100-percent) scale. Zero overflow values are currently colored light blue.

End Chain Overflows — This coloring is similar to the *Overflow* coloring described above, except that it is based on a special overflow peg count in the CCSIM switch reports that counts only when the overflow occurs on an attempt for which the pegged trunk group was the end of the routing chain. Such counts correspond to blocked calls. Consequently, *End Chain Overflows* are much more useful in identifying problem areas in the network than are ordinary *Overflows* which tend to show high values in busy-hour traffic even though no calls are blocking. This peg count is not available in the current real DSN.

ACH — Attempts per Circuit per Hour (*ACH*) is computed by combining the attempts to find a free trunk for all precedences, dividing by the number of working trunks, and multiplying by 12 to change the 5-min report data to hourly values. The resulting value has no sharply defined upper limit. We use a green-to-red coloring ranging from 0 to 2400 with values in excess being truncated to 2400. The 2400 top was determined empirically from watching some DSN-Pacific simulation runs with damage scenarios. It is easily changed if future work shows that a different value would be more useful.

Holding Time — An estimate of average call holding time that is computed by dividing total usage by the sum of the incoming attempts peg count and the outgoing attempts less overflows. The resulting number is translated to a green-to-red scale with full green at 5 min or longer and full red at 20 s or shorter. All precedences are combined for this coloring. Short holding times (red) indicate that trunk resources are not being used effectively.

OCCH — Outgoing Connections per Circuit per Hour (*OCCH*) is computed by subtracting the overflows from the attempts and dividing the result by the number of working trunks in the group and multiplying by 12 to get an hourly value. The color is translated to a green-to-red scale with full green at 15 or less and full red at 115 or more. The limits were empirically worked out, and are readily changed if indicated.

ICCH — Incoming Connections per Circuit per Hour (*ICCH*) is computed by dividing the incoming attempts by the number of working trunks as is displayed using the same scale as *OCCH*. We find it useful in indicating the destination of a focused overload traffic pattern in which there are many attempts to reach a few stations so that most attempts reach a busy destination.

Any desired portion of the map display may be expanded to fill the entire map window by holding down the left mouse button and moving the mouse so that a rectangle generated by the program encloses the area to be magnified. The starting point will be one corner of the rectangle. The point at which the button is lifted will become the diagonally opposite corner. Lifting the button causes the magnified view to appear. The link widths as well as the rectangles corresponding to switches are magnified in the expanded view, as shown in Figure 2-2. The increased link width makes it easier to perceive subtle differences in link coloring. Links that cross the rectangle in the original map, but do not terminate on a switch within the expanded view, are not displayed in the expansion. The expansion may be carried further by repeating the process within an expanded view. Links going off the expanded view are labeled with the destination switch name but, in a very dense grouping of links, the labels are sometimes obscured by overlying links. Further work is indicated in this area. To return to the full network map, the user holds down the left mouse button and moves the mouse off the edge of the window.

The link colorings are currently the only means by which network state information is presented to the interactive user. In the upcoming year, we expect to add node colorings to show switch data and numerical presentation of both trunk and link data on request by mouse interaction.

To get data on network performance as seen by the simulated users, the experimenter can request displays of Grade of Service, Offered Calls, Blocked Calls, Call Intentions, Failed Calls, Call Failure Rate, and Mean Tries for Success. These statistics are obtained using the MATRIX OPERATIONS action option with the DISPLAY STATISTICS sub-option. The menu brought up in the lower right window allows the selection of which particular statistic is to be displayed. The desired precedence level may also be selected, and a source or destination node may be specified. If neither source nor destination is specified, a numerical presentation of the full point-to-point statistics matrix is presented, replacing the network map in the large window. For a large network, the matrix will be bigger than the window, but the window is made scrollable so that the user can move the matrix to view any part in the window. Pushing any mouse button with the mouse positioned on the matrix will cause the network map to reappear.

In the statistics display mode, if a source or destination node is selected [either by typing the node name or moving the mouse to the node and pushing the left (source) or middle (destination) button], the display will show the appropriate row or column of the matrix, placing the numerical information above the associated switches. This mode provides a means of showing graphically which nodes are having trouble getting to or hearing from a particular node.

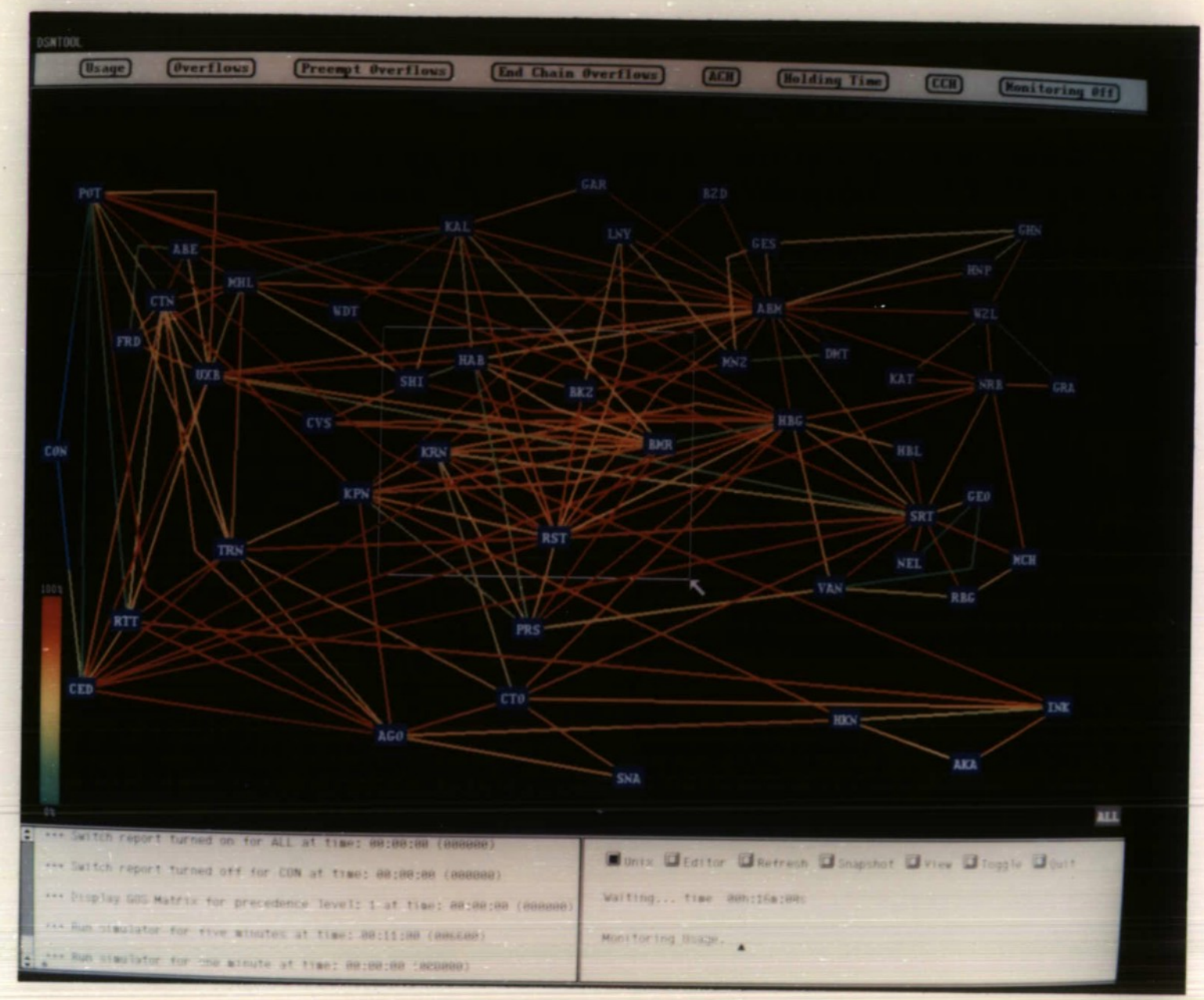


Figure 2-2. Photograph of IDSIM graphics interface display showing expansion of a portion of the European DSN network. Link coloring corresponds to trunk usage.

When the lower right window of the display is not being used to get parameters for a command, it contains a set of mouse buttons that offer access to capabilities not directly associated with running a simulation. The buttons are labeled as follows:

Unix — This button provides access to a Unix command window that can be used for any Unix activity that an experimenter might want to carry out during an experiment. Use of this feature does not stop a simulation, but it may slow the progress by making demands for CPU and memory resources in the workstation.

Editor — This button changes the mode of operation of the graphics interface. It brings up a new window along the bottom of the screen which has a number of buttons which support editing as well as original entry of the network map. By using this capability, switches and links can be added to or removed from the map. Switches can be moved, dragging their associated link ends with them, and links can be bent in one or two places to prevent them crossing over a switch or to avoid crowding. The Editor is not normally used during a simulation run, but it can be; and, if it is, the simulation will continue until another "run" command from the interface is needed. From the Editor mode the user can cause the edited network map to be saved if desired.

Refresh — This button causes the network map and its link colorings to be redisplayed. It is needed to get the correct link coloring for the links attached to a damaged switch. Such colorings are normally updated only when switch reports are received, but none come in from a damaged switch, so the coloring corresponding to the last report will remain indefinitely. Refresh is also useful to remove the numerical statistics display from the network map when the source/destination statistics mode has been used.

Snapshot — Snapshot causes the switch report data corresponding to the current display to be written to a file with a name requested from the user. We plan to augment this feature by adding the complete set of simulator statistics to the file so that the snapshot will be useful in associating a network state (switch reports) with a performance assessment (user statistics).

View — View requests a name for a snapshot file which is read in and used to color the links in the network map. Of course, the snapshot must correspond to the network being displayed or an error message will be generated. With the augmentation of the snapshot to include the statistics matrices, the graphics interface can serve as a stand-alone facility for scanning data accumulated over a series of experiments.

Toggle — Toggle provides the means of switching between two network colorings, either the current simulation results and a snapshot retrieved via the View button, or two such snapshots. We do not anticipate extending this capability beyond two sets of data because the computer memory needs for each set of data are quite large, and swapping this amount of data to and from the hard disk slows down the operation substantially.

Quit — This button provides the mechanism for shutting down CCSIM and exiting from IDSIM to the underlying Unix environment of the workstation. A menu asks the user if the exit is really wanted, and whether or not the summary statistics from CCSIM are to be generated when it shuts down. In many situations, those statistics are not of interest since they combine a number of different situations that were explored during the run. Use of the Quit button also stops NMES if it is running as a component of IDSIM.

2.2.2. Command Files

Command files constitute an alternative mechanism for carrying out a network simulation experiment. By interspersing "time" commands with commands for network traffic changes, node and link damage, and switch control application and removal, complete experiment scenarios can be carried out without human intervention. With the exception of creating snapshot files, one can do anything from a command file that can be done from the graphics interface. Eventually, we intend to support the snapshot capability from command files, but this work presently does not have high priority.

As noted earlier, in the past year we have removed command file processing from CCSIM itself and have created a new program to perform that function as a separate component of IDSIM. The separation simplifies the communication between CCSIM and its client processes, and assures that all CCSIM functions can be controlled in a uniform way by the other IDSIM components.

2.3. NETWORK MANAGEMENT EXPERT SYSTEM (NMES)

The other major component of IDSIM is the enhanced NMES. Subtask IB of the SOW specified the expansion of the NMES knowledge base, development of an interactive interface with CCSIM, and development of operating features providing for convenient use of IDSIM for network management investigations. These goals have been realized in the course of the year, as described in the following paragraphs.

In the FY87 Annual Report, we described a prototype expert system developed on a SUN 3/260 workstation using SUN Common LISP and the ART expert system development shell (Figure 2-3). This has formed the foundation of our FY88 efforts to develop the enhanced NMES. Briefly reviewing the prototype expert system architecture, at the lowest level the network representation and switch reports are received and processed into the ART frame representation for object descriptions, called schemata. These schemata are stored in the ART database and are scanned by many specific monitors for interesting features and information. The outputs of the monitors are stored in the database as the abstract state of the network. Potential problems are identified by a module which looks for patterns in the abstract state of the network, and these problems are confirmed over time by a confirmation module. A planning module recognizes a problem and devises control actions to improve or correct the situation. An observation module observes the effects of the applied controls over time. Each module has the ability to "turn on and off" monitors. For example, once a problem is recognized and controls are applied, monitors watching for the problem to return to normal are "turned on." NMES also has its own graphics interface that displays its observations and conclusions and allows the experimenter to turn monitors on and off and to adjust some monitor threshold parameters.

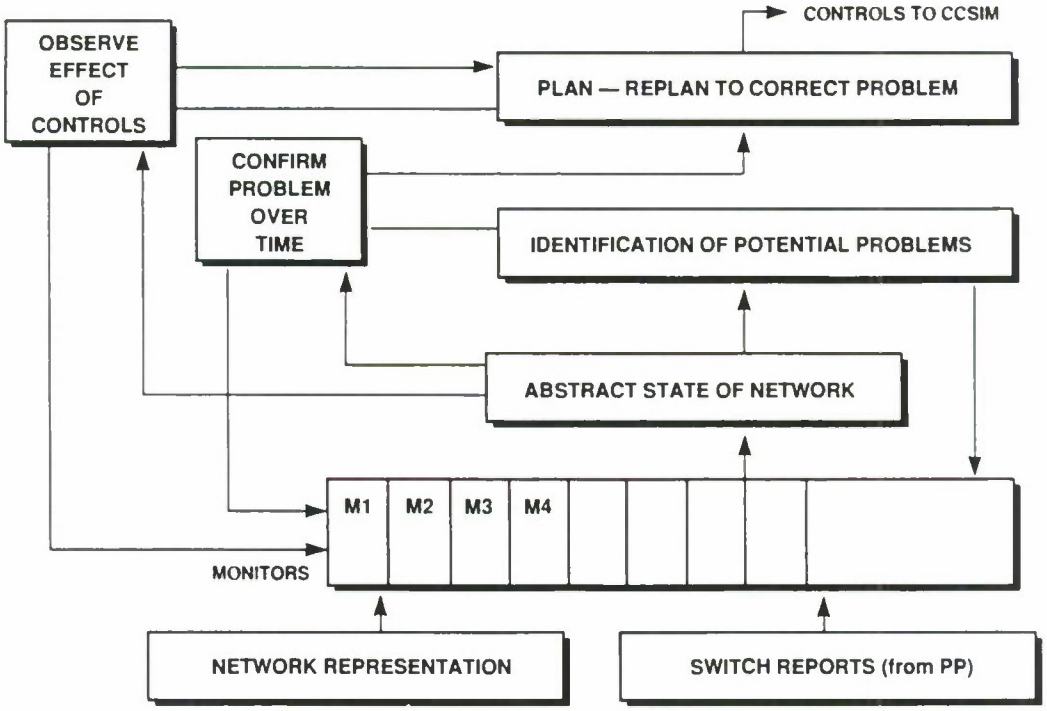


Figure 2-3. Expert system structure.

Starting with this prototype architecture, during FY88 the expert system was developed into a fully integrated component of IDSIM capable of receiving, processing, and analyzing switch report information from CCSIM and sending network management controls back to it. NMES has progressed to include many additional monitors, a problem detection module, a confirmation module, and a planning module. The additions to the functional architecture have centered around the detection of damaged nodes, restored nodes, overloaded trunk groups, trunk groups that are no longer overloaded, damaged trunks, and restored trunks. These additions are discussed in more detail in later paragraphs of this section and in Section 3. We expect that these NMES capabilities, as a minimum, will be demonstrated at DCA-Europe in 4Q FY89.

One of the fundamental developments necessary for a fully integrated NMES was a communications interface capable of sending and receiving commands from CCSIM. The present NMES communications interface consists of a number of routines written in the C programming language and accessed from LISP as foreign function calls. These routines set up pipes in a manner similar to the communications between CCSIM and the graphics interface. NMES receives switch reports and time information from, and sends network management controls to, CCSIM through these pipes. Both NMES and CCSIM run faster than real time, but, in general, CCSIM runs faster than NMES. To avoid a backup of switch reports sent by CCSIM and not yet processed by NMES, and to avoid inconsistencies between the time the expert system applies control commands and the time CCSIM receives them, NMES allows CCSIM

to run in 5-min increments during which NMES receives and processes switch reports. When CCSIM stops at the end of the increment, NMES calls the ART shell to determine whether any rules in the knowledge base are applicable in the current situation. Any such rules will be "fired" and appropriate control actions will be sent to CCSIM, database updates will be carried out, etc. At this point, the NMES graphics display is updated to show any conclusions NMES has reached. Any control actions sent to CCSIM are also recorded in the logging window of the CCSIM graphics interface and the log file kept by CCSIM. NMES then sends a message to CCSIM to run for another 5 min. CCSIM will then continue unless it has reached a stopping point set by the user through the graphics interface or command file.

The lowest level of NMES is made up of "monitors" which continually scan switch information received from the network for patterns and features of interest. These monitors reduce a large volume of switch information to a small number of interesting or unusual facts. The monitors presently implemented are listed and defined in Table 2-2. The outputs of the monitors listed in the table are stored in the abstract state of the network database. The user is able to see the current activity of any monitors by using the mouse to turn the display of selected monitors on and off. The expert system has a more important type of control of the monitors: it is able to control which monitors are "turned on" at any point in time, i.e., which patterns it wishes to search for in the switch reports. For example, in the normal state of the network a node will be responding; therefore, there is no reason to recognize nodes responding. If a node has been declared damaged and appropriate control actions have been taken, the network must be monitored to determine if the node has been restored. Controls such as CB (code block) applied to the net in a damaged node situation will do more harm than good if they are left in effect after the node is restored, so they must be removed as soon as possible after restoration. So far, the only monitors that are "turned on and off" are ones to determine that a problem has been corrected, but there will be other uses for this "on/off switch" in the future. For example, in order to locate overloading resulting from a failed node, the expert system might turn on overload monitors in certain areas. Thus, monitors could be turned on to look for patterns of overloaded trunk groups within and leaving an area when the primary switch in the region has been damaged.

Some monitors, such as the one to locate trunk groups that are overloaded, have thresholds that can be varied by the user through the NMES graphics display. For example, in the lower right-hand corner of the display the user is able to enter the percentage that must be exceeded by the ratio of end-of-chain overflows to access attempts in order for an overloaded trunk group to be declared. The ability to vary this threshold is helpful in carrying out knowledge engineering experiments.

At higher levels in the expert system functional architecture, modules identify potential problems and confirm them over time. For example, a node is considered to be damaged if its neighbors see no incoming signals from it and no switch report is received from it during one time period. In this case, meeting the conditions for one 5-min time period is deemed sufficient for the conclusion, but in other cases longer time periods are needed. The rule that identifies an overloaded trunk group currently requires that end-of-chain overflows occur on the group, and that the ratio of the number of overflows to the number of access attempts exceeds a threshold for N successive report periods. The outputs of these modules are stored in the database.

TABLE 2-2
NMES Monitors Currently Implemented

node-not-responding	Finds any node that has not sent a switch report in the last 5 min
node-responding	Finds any node that was earlier declared "down" by NMES, and has just transmitted a switch report
neighbors-see-no-incoming-signals	Finds any node whose neighbors all report no incoming call attempts from that node
neighbors-see-incoming-signals	Finds any node that was earlier declared "down" by NMES, and has just made one or more call attempts to its neighbors
node-where-calls-fail	Finds any node reporting call failure events for originating calls, tandem calls, or both call types
overflowed-trunks	Finds any trunk group reporting call overflow events
overflowed-after-preemption	Finds any trunk group reporting calls that attempted preemption on that group and overflowed nonetheless
end-of-chain-overflow	Finds any trunk group reporting calls that had no routing table options left after the current attempt on that group, and overflowed in the attempt
trunks-overloaded	Finds any trunk group that has reported a ratio of overflows to attempts above a threshold
overflowed-low-usage-trunks	Finds any trunk that has reported overflows on trunks with low usage

The planning module analyzes the database looking for situations that it is able to correct or improve by devising a set of control actions to apply to the network. When it recognizes a problem that it can ameliorate, it sends controls to the simulator and turns on monitors to determine if the situation changed so that the controls could be removed or altered. Presently, damaged nodes and overloaded trunk groups are problems for which NMES is able to devise a set of controls. When a node is determined to be damaged, the planning module generates commands to CCSIM that will apply CB controls at all other nodes prohibiting any calls to the damaged node, as well as SK (skip) controls prohibiting routing on any trunk groups from neighboring nodes to the damaged node.

The development of knowledge about effective control actions to compensate for overloaded trunk groups is still in a preliminary stage. A control action which slightly improves the overloaded-trunks situation has been devised, but further experimentation will be necessary to determine what controls (if

any) will be really effective. The present candidate control action is to cancel calls overflowing the overloaded trunk group by means of a 1-percent CANF (CANcel-From) control with a recorded emergency announcement option which is assumed to convince the caller to give up the attempt. This control action realizes a slight improvement in the GOS. A greater improvement in GOS could be obtained by canceling a greater percentage of the calls, but experiments to date indicate that doing so will often increase the overall call failure rate.

During the past year, we have begun to use the expert system as a tool for knowledge engineering. Thus far, such use has focused primarily on tuning monitor parameters and planning rules related to overloaded trunks. In a lower panel of the expert system display, the user is given the ability to vary the overload threshold (the value that must be exceeded by the ratio of end-of-chain overflows to call attempts, in order for a trunk to be declared overloaded) and the CANF percentage (the percentage of calls canceled by the CANF control applied when a trunk is overloaded). Being able to vary these two parameters conveniently has facilitated rapid exploration of wide ranges of parameter settings in this tuning process. Our best results so far have been obtained by canceling 1 percent of the calls overflowing from a trunk group that was at least 75 percent overloaded. This action typically improves the call failure rate for routine calls by no more than a couple of tenths of a percentage point. Precedence calls are not affected, since no precedence-call blocking is occurring under the assumed scenarios.

The tuning of other NMES components is discussed in Section 3 of this report which addresses simulation studies and network management issues.

For the DCOSS controller trainer and NMES demonstrations discussed in Sections 4 and 5, we will be making a number of changes and enhancements to NMES in FY89. One will be a shift from a link-oriented view of the network to one that deals with individual trunk groups, of which there may be many between pairs of switches. This is necessary because DMS switches send statistics reports, and implement control commands, on a per-trunk-group basis rather than the per-link basis we have used in CCSIM and NMES until now. The volume of switch report data will increase with more trunk groups reporting, and NMES will need to supply the group name on all controls sent to switches. We believe this change can be accommodated by NMES since it involves no fundamental redesign of the database or knowledge base of rules.

Another area in which we are planning changes is the NMES user interface. We have found that, although it requires two SUN workstations to run both NMSS and CCSIM, it would be convenient to be able to view both the simulator state and the expert system state from one display monitor. In fact, it can be quite difficult to run experiments involving both NMES and the CCSIM graphics display, because of the need to have two SUN monitors in one room or to run between two rooms to access the two interfaces. We have also found that the SunView graphics available through LISP macros is unsupported by SUN and is incomplete with respect to the C-based SunView graphics. A new expert system interface will therefore be written, as an independent process coded in C and communicating with NMES in a manner similar to the way CCSIM communicates with its graphics interface. One of the side benefits of this approach is that NMES will be able to use features such as zooming, presently used in CCSIM graphics, without translating them into LISP.

3. SIMULATION STUDIES

Task II in the FY88 Statement of Work provided for performing traffic overload/congestion/damage simulation studies with IDSIM. One purpose of these studies is to develop knowledge of switch statistics report patterns corresponding to classes of such network problems, leading to the ability to recognize incipient problems before conditions become critical. The other main purpose of the simulation studies is to learn how to respond effectively to these problem categories by selecting and applying NM control actions. Both kinds of knowledge are steadily being added to the knowledge base of NMES through implementation of new monitors and rules, as described in Section 2 of this report. FY88 accomplishments in the simulation studies area have included: considerable further scrutiny of the ramifications of switch damage scenarios and appropriate control actions; evaluation of the use of the CANcel-To (CANT) control as a response to switch outages; experiments with ReRoute (RR) to improve call failure rate under certain switch outage scenarios; studies of the damaged trunk group and noisy trunk group problems; studies of two kinds of focused overload conditions (high overall traffic to/from a switch or region and high traffic to a few busy destinations); and studies of detection and correction of nonfocused trunk group overload conditions.

The near-term goal of the work reported in this section is to gain understanding of the NM problems to be expected in the evolving DSN, both to support building effective strategies into NMES and to provide recommendations for DCOSS operator control actions that might be undertaken in the real DSN in the period before a complete network management capability is realized. Working toward this goal, we have been extending the results reported in our 1987 Annual Report* by carrying out simulation experiments using IDSIM as well as talking with people experienced in network management of commercial telephone networks and studying reports such as the Network Management Handbook for NMSS, the Network Management Support System created by GTE for DCA-Europe. This report (GTE Report 00-1900674, dated 30 June 1987) gives useful insights into the thinking of experienced network controllers.

In the following paragraphs we present our conclusions to date on a number of NM issues as indicated by the subsection titles.

3.1. SWITCH DAMAGE

The first area that we examined in our NM experiments was single-switch damage. We reported on the results of our investigation of a number of scenarios of this type in the previous Annual Report. There are two aspects to the problem. One is to recognize that a switch is not functioning, i.e., not switching calls. The other is to determine what control actions, if any, can reduce the impact of the switch loss on calls that do not require the switch to be up in order to reach their destinations. In our work to date, we have used the rule that if a switch does not report, and all its neighbors report seeing

* Annual Report, "Knowledge-Based System Analysis and Control Defense Switched Network Task Areas," Lincoln Laboratory, MIT (30 September 1987).

no incoming traffic from that switch for a complete reporting period, then the switch must be down. If all the neighbor nodes are in fact reporting, and if the current traffic from the suspect switch is expected to be well above zero, this conclusion is quite likely to be valid. However, if some switches do not report, or the traffic at the moment is very low, the conclusion may be incorrect. For the demonstration discussed in Section 5, NMES will not get reports from some of the neighbor switches because they do not report to the NMSS to which we will be connected. In such a case, it will be necessary to know the traffic density as a function of the time of day in order to evaluate the probability that the absence of reported traffic from the suspect switch may be normal for the time of observation. If the expected traffic at the time is quite low, the decision should be delayed to allow more observations from the reporting switches.

A potential alternative means of identifying some switch outages in the NMSS environment is to conduct a series of simulations in which one switch at a time is disabled, and to look for recognizable patterns in the statistics reports from the switches that are connected to the NMSS. Besides providing corroborating evidence for a suspected outage of a switch on the NMSS, this option may allow identification of outages in certain switches elsewhere in the net but close topologically to NMSS switches.

The second aspect of the switch damage problem is working out a plan for ameliorating the effects of the damage. NMES currently calls for CB (code block) controls to all destinations reachable only through the damaged switch, plus SK (skip) controls for the trunks from neighbor switches to the damaged one. The CBs remove traffic (both routine and precedence) that cannot succeed from the network. The SKs prevent routine and precedence traffic for other destinations from attempting to route through the damaged switch. For the Pacific DSN networks we have simulated, this combination of controls appears to be a quite effective first-order fix for the problem, and we recommend such action as a standard initial response. The CBs prevent precedence calls to the damaged node from preempting unnecessarily, thereby removing one of the most damaging effects of the switch loss. Because routing in the Pacific is quite generous in providing alternate routes via other trunk groups and switches, precedence traffic to destinations other than the damaged node is nonblocking in the cases we have examined; moreover, the SKs give the routine traffic a good chance of completing successfully. Statistics from CCSIM show that for single-switch damage there are virtually no failures of precedence calls above routine, and that there is almost always some capability for communication among all pairs of switches for routine traffic. As would be expected, the statistics also show that calls between some switch pairs have a much lower probability of success than others. Further control actions can help with this fairness issue. We present an example of such control action later in this subsection.

In our simulations of DSN-Europe single-switch damage scenarios, we found that the CB and SK prescription was not effective in assuring the completion of precedence calls to and from undamaged switches. We had seen similar problems with precedence calls in multiple-switch damage scenarios in DSN-Pacific simulations and were able to correct the problems with routing changes. Routing in Europe is much more restrictive than it is in the Pacific with respect to the availability of alternate routes for tandem calls. When we first ran such scenarios and observed the effects, we questioned whether our

engineered routing algorithm and the tables we had been given by DCEC constituted a correct representation of DSN-EUR routing. We have made inquiries and have been told that they are probably correct and that the effects we observe are to be expected. It would be reassuring to obtain additional corroboration of the correctness of our simulation by conducting a comparison such as that described in Section 2.1.2 but using DSN-EUR topology and parameters, and we plan to discuss this possibility with DCEC. We expect that routing changes would be effective for assuring completion of precedence calls in the European net as well, and we plan to carry out further experiments to test that expectation.

In our simulations, control applications are instantaneous. In the real network, however, they can take a relatively long time, particularly in the current situation where there is no direct computer-supported access to the switches. The NMSS Handbook estimates 3 to 4 h to apply the full set of CBs required to deal with the loss of a major node. In the meantime, the Handbook suggests the use of CANT controls on the trunks to the damaged nodes to cancel calls that cannot succeed because they are being routed through the damaged switch. If the canceled calls get an appropriate Emergency Announcement that discourages retry, the control action will have some of the effect of the CBs. The difference is that the calls will tie up some network resources getting to the switch at which the CANT controls are applied. This action contrasts with our use of SK controls on the trunks to the damaged switch. The SKs will cause call attempts to fail if there are no other routing options, but will not discourage retry because the blocked calls will not get an Emergency Announcement. They will, however, allow some calls to succeed that would be canceled by the CANT controls, and our experiments showed that SKs are to be preferred in DSN-Pacific where routing is robust. We have not yet done comparable experiments for DSN-Europe.

Going beyond the CB plus SK prescription, we seek NM actions that can improve service for the remaining reachable users. It has been our observation that, except in a few special situations where the network topology may allow SK controls to approximate a reroute, it is necessary to change the routing tables in the network to achieve any consequential improvement in performance. As an example, we tried a scenario in which we damaged PRL in the Pacific DSN using the 1989-90 23-node network and looking at the behavior with Western Pacific busy-hour traffic. With PRL gone, all traffic in and out of the Hawaiian region must go through the remaining gateway switch, SCS. After applying the indicated CBs and SKs, we observe that even though the overall call failure rate (CFR) is only 2.7 percent (GOS = 0.378), traffic from Hawaii to the southwest (Philippines, Okinawa, and Guam) is experiencing a 12.7-percent CFR, and traffic from the southwest to Hawaii is faring even less well with a 24.2-percent CFR. Corresponding GOSs are 0.661 and 0.780, respectively. Switch reports show that trunking from SCS to the southwest is fully occupied.

After studying the routing tables and traffic statistics, we concluded that the situation might be improved by diverting traffic between the SW Pacific and CONUS through Japan. The original routing table used any direct route first, then alternate routes through Hawaii, and finally alternate routes through Japan for such traffic. We simply switched the order of trying the alternatives. We also observed that, with PRL gone, we could allow more alternate routes to be used without introducing routing loops. We edited a new routing table with the indicated changes, and ran a simulation that switched to the new table at damage time. The results showed an insignificant improvement in overall CFR to 2.0 percent, but for

traffic from Hawaii to the southwest we now observed a much better 5.6-percent CFR and somewhat better 17.9-percent CFR in the other direction. While these results showed a considerable improvement, there was still a bias against traffic headed toward the Hawaiian region. To deal with this, we tried directionalizing the trunks to SCS. By using the DRE control (see Appendix A) to reserve one trunk each for incoming calls on the links to SCS from CKS (Philippines) and from FBK (Okinawa) in conjunction with the routing change, we observed an 11.9-percent CFR from Hawaii to the southwest and a 10-percent CFR in the reverse direction. The GOS observed by these callers (0.715 and 0.685) was still very high, but roughly 90 percent of their calls succeeded due to retries. It should be noted that the DRE control is very sensitive in situations where the trunk group size is not large (40 from CKS, 39 from FBK). Reserving just one trunk on each of two links was enough to correct a roughly 3-to-1 imbalance in CFR.

The control actions used in the scenario described in the previous two paragraphs improved performance only for routine traffic, but they would have been equally effective for precedence traffic if the level of such traffic had been high enough to overload the original routes.

It may be possible to make further improvements to the routing in this damage situation. We make no claims for optimality for the results presented here. It should be possible to run algorithms such as are used in network design to get the best routing tables for the remaining resources and expected traffic patterns. NMES may be able to carry out such computations in the future, but it is not clear that the results would be useful. The only direct control that network managers have over routing is to select among some set of prestored tables by sending REROUTE controls to switches which support that feature. Any fundamental reworking of the tables would require manual operations in each switch to set up the tables followed by a switchover that might be effected by REROUTE controls. Such an operation would presumably be needed to reconstitute the network from surviving segments in a post-attack situation.

We have not done many experiments with routing changes because such experiments in the current CCSIM involve creating a hand-edited routing table for each situation. Only one routing-table change can be made for each simulator run. The editing process is slow, tedious, and prone to error, particularly for large networks such as DSN-Europe where the very long lines of text wrap around the workstation display screen making it difficult to ascertain exactly which entry is being changed. In FY89, we plan to change CCSIM to allow an unlimited number of routing-table changes during an experiment and to provide editing and validity checking aids for building new tables.

It should be noted that in the scenario described above we used information from the CCSIM performance statistics to decide which source-destination pairs were experiencing difficulty in communicating. This information is not available to a network manager in the real-time switch report data he gets from the switches. He can observe which switches are experiencing blocking on attempts to route calls and which trunks are overflowing, but he can only guess at the sources and destinations of the failing calls. In dealing with such a situation, NMES will have to make assumptions about the traffic pattern; for example, devise a plan for changes by using time of day to look up an expected pattern, and by working with that and the known current routing table. If an algorithmic approach proves too cumbersome, then NMES can fall back on a set of cookbook actions based on human study of many scenarios

with IDSIM. Such a procedure is all that human controllers would be likely to be able to carry out. The more difficult problem of reconstitution after major damage would require involvement by persons with network design experience.

3.2. TRUNK PROBLEMS

We have addressed two types of trunk problems in our studies. The first is partial or complete loss of a trunk group. The second is deterioration of trunk quality to a point that users find the connection unsatisfactory, hang up, and retry their call. We will refer to the latter type as the "noisy" trunk problem, and the former as the "damaged" trunk problem. We do not yet simulate the latter type in CCSIM, but we expect to add the capability in the future.

Damaged trunks are detected by the associated switches when they attempt to use them. Our model of switch behavior assumes that a switch which has detected a bad trunk either by a failure to handle in-band signaling or on a transmission check during a call setup will not attempt to use the trunk again until it is informed that the trunk is back in service. Whether or not all switches will report the number of working trunks in a group is not too clear to us. The Northern Telecom DMS switches do apparently report that number, so there is no problem in recognizing trunk damage for such switches. On the assumption that there may be switches that do not report the number of working trunks, we did some experiments to determine the extent to which we could ascertain that number from other reported data. We reasoned that if a group was showing overflows,then its usage reports normalized with respect to its nominal group size should show approximately 100-percent values. If smaller values were observed together with significant overflow percentages, we could deduce that the number of working trunks had gone down by roughly the same ratio as the observed normalized usage had to 100 percent.

We gathered some statistics for usage and overflows for a range of trunk group sizes, and concluded that we could estimate the number of working trunks in a group from these data and the nominal group size provided we had 10 percent or more overflows from the group and normal call holding times. The accuracy of the estimate improved as the overflow percentage increased, and became less reliable when the number of working trunks left in the group became small. Using this information, we wrote and installed in NMES a monitor that locates trunk groups which have experienced both overflows and low usage during a time period. This monitor can be turned on for switches not offering information about the number of working trunks in the group. We have not done any fine tuning of this monitor, since we do not now expect to have such switches in DSN-Pacific or the subset of DSN-Europe that we will be dealing with in our field demonstration.

Unlike the situation with damaged switches where CBs and SKs seem to be uniformly helpful, there does not appear to be any simple set of control actions that should automatically be undertaken. Loss of trunk capacity can be a more serious event than loss of a switch because there is no compensating loss of offered traffic to the net. In some of the scenarios we have examined, routing changes were helpful in improving network performance, but it can be quite challenging to figure out what changes to make. In many cases with DSN-Pacific simulations, we find that no action at all is needed. The many alternate routes are able to cope with the outage for normal traffic situations. We have not tried trunk outages in

DSN-Europe simulations, but we expect that reroutes will clearly be needed for such situations just as they are needed for switch outages. When the only routing option involves using a damaged trunk, the only fix is to change the routing.

For trunk damage as well as other losses of network capacity, there are cases where it would be desirable to force some traffic to use alternate routes further down the routing chain, thereby spreading traffic over more routes and reducing the traffic offered to a damaged trunk or congested switch. Such behavior would be realized by a "spray" routing option which would distribute the traffic over the set of alternatives, rather than moving to an alternate only when all earlier routes in the chain have overflowed. This effect can be approximated by the use of fractional SK controls, and we have used this technique successfully in some scenarios, but it has the problem that SK affects all traffic on a trunk, not just traffic headed for a particular destination. When most of the traffic on a link is the traffic to be sprayed, SK works well. Otherwise, some of the diverted traffic may be lost because it skips off the end of its routing chain, which is different from the chain for the destination for which the spray effect was intended.

For the noisy trunk case, the symptom is unusually short call holding times. This symptom by itself does not uniquely indicate a noisy trunk group; holding times will also be short in focused overload situations, and in cases where precedence calls are failing to complete due to trunk or switch damage. Consequently, those cases should be recognized and dealt with before looking for noisy trunk symptoms. If there are multiple parallel trunk groups, and short holding times are observed on one but not on the others, the likelihood of a noisy trunk problem is increased. The indicated NM action for a noisy trunk group is to apply a SK control on it and request a maintenance check. We plan to build a monitor for NMES to watch for low holding times on trunk groups and recognition rules to check for the noisy trunk case in the field demonstration environment. We think it may be more likely to occur than a real outage of a switch or trunk.

3.3. SWITCH CONGESTION

Switch congestion occurs when the switch CPU cannot keep up with the call processing load. The situation may be aggravated by failure or inadequate provisioning of the switch peripheral devices that service trunks and subscriber lines. The symptoms of congestion are slowness in providing dial tone to users and in responding to signaling from neighbor switches. The latter effect causes congestion to spread to some degree by slowing call processing in the neighbor switches.

Because trunking is relatively thin in relation to switch processing power in the DSN, switch congestion should not occur at any level of interswitch traffic if such traffic were the only load. Barring exotic failure modes of the switches, we expect that congestion would only result from subscribers attempting calls at too high a rate. The remedy for this situation is to deny dial tone to some users by means of a NM control called LINE LOAD CONTROL (LLC). This control can be activated either automatically when CPU usage exceeds a threshold, or manually by NM personnel. We have no data on the time required for this control action to bring the congestion under control, but we expect that it would be measured in minutes rather than hours. If the time is short, there would seem to be little need for

other NM action to divert traffic around the congestion. If the time can be long, however, approaches such as those discussed above for rerouting around the congestion could be used.

CCSIM does not currently simulate switch congestion, but we need to add that feature in some form for the trainer application described in Section 4. It is not practical to simulate in detail the local traffic needed to produce switch congestion. It would make CCSIM run too slowly to be useful. We expect to model the effects of local traffic using aggregated traffic values that can vary with time as input, and generating congestion symptoms as output. Our problem is to get data on the dynamic behavior of the congested switch in response to LLC application so that we can make a realistic simulation. With such a capability in place, we will be able to carry out experiments exploring the need for other NM actions in response to switch congestion.

3.4. FOCUSED OVERLOADS

There are two types of focused overload situations that can be explored with IDSIM. One is simply a general increase in traffic beyond busy-hour levels into or out of a switch or region. This case is readily simulated using the LOAD-LEVEL command to CCSIM described in Section 2.1.1. Our experiments with scenarios of this type do not show any very unusual effects. Trunks overflow, and precedence calls may be lost if the traffic excess is large enough so that trunk resources are overloaded with precedence calls. Partial CBs are effective in reducing the stress on network resources so long as the overload does not reach a point where precedence calls are failing. We have found no prescription for the latter case.

The second type of focused overload occurs when there are many callers to a particular destination code or small set of codes such that most of the callers reach busy destinations. This situation results in short holding times on the trunks involved, with a consequent inefficient use of trunk resources. To model this in CCSIM, we created a fictitious end office attached to a backbone switch with a small trunk group, and then using LOAD-LEVEL increased the traffic to this destination well beyond the capacity of the trunk group. Most of the traffic blocked in the backbone switch, creating very nearly the same effect on the network as would be created if the calls had reached busy destinations. This failed traffic can include precedence calls that are blocked by equal or higher precedence calls already in progress to the overloaded destination(s). In this case, we observed short holding times on many trunks in the network, and by comparing the values for ICCH (Incoming Connections per Circuit per Hour) with OCCH (Outgoing Connections) we could easily see what backbone switch was the target of the overload. A fractional CB to the fictitious end office (which applies only to routine traffic if the fraction is less than 100 percent) readily removes the symptoms, except that it cannot help the precedence call blocking possibility just mentioned. The fraction of routine traffic to block depends on the extent of the overload and can easily be adjusted in IDSIM, but not so easily in the real net because of the time required to apply or adjust the CBs. If hours were required to get the controls into place, it seems unlikely that the effort would be worthwhile, because the situation causing the overload would probably have changed or disappeared.

Another problem faced by the network manager in dealing with a focused overload of this type is that, if the overloaded destination codes are in a large office, he does not know what codes should be

blocked unless he has information from some other source. The switch reports tell nothing about which codes are causing the problem. In commercial networks, the managers depend on the media to warn them of rock concert ticket sale openings, radio show giveaways, etc. that produce such overload events. They then know which codes to block. We do not know whether events of this type are likely to occur in the DSN or how, if they do, the managers will find out which codes to block. Fractional blocking of all codes to the destination office will alleviate the symptoms, but will block many calls that could have succeeded; moreover, it will be ineffective in reducing any precedence call blocking that may be occurring because of precedence calls already in progress to the desired destinations.

We are implementing monitors for NMES to look for the symptoms of a focused overload to appear and to disappear after CB action. Having detected a focused overload, NMES will have to ask the operator using it to provide information about the codes that should be blocked. It should then be able to implement the CBs and adjust the fraction to be blocked in a feedback fashion until the problem disappears.

3.5. OVERLOADED TRUNKS

In the 1987 Annual Report, we described switch damage scenarios in which we were able to improve the grade of service for users who can still communicate after the damage by canceling some calls with an Emergency Announcement that discouraged retrying. In CCSIM, such a canceled call is not retried. In the real network, some probably would be retried; if so, the results we get (which are not all that good anyway) should be viewed as optimistic. We generally find that we can cancel a few percent of the calls that overflow from overloaded trunk groups, and get a small improvement in GOS without seriously increasing the call failure rate. In some cases, we see a small improvement in CFR as well. The theory is that if we could cancel calls that cannot succeed in spite of retrying, then we could remove a number of call attempts from the system without causing any additional failures. Those wasted attempts would release some trunk resources that might allow a few more calls to succeed. The problem, of course, is that we cannot easily identify calls that cannot succeed, and the strategy involves a risk of canceling calls that could have succeeded, thereby worsening system performance.

The problem of finding the right calls to cancel is particularly difficult in a network that uses the kind of engineered routing we find in DSN-Pacific. In a hierarchical routing scheme, some trunk groups are designated "high usage" and are the first routing choice, while others called "final" are the last in the routing chain. If we cancel calls overflowing from final groups, we have a relatively good chance of canceling a call that cannot succeed. At least such a call is being blocked on the attempt at which it overflowed. In most cases in DSN-Pacific, there is no such thing as a final group. A group will be "final" for some routing destinations, and the first or second choice for others. In such a case, there will be many overflows from the group, only a few of which are good candidates for cancellation. We therefore have to play a statistics game, using the control at a very low setting (typically 1 percent) and counting on the fact that call intentions which are having difficulty and retrying many times will come by the control more often and have a higher probability of being canceled than those we do not wish to cancel. The result is typically a small change, and it affects only routine traffic; the precedence traffic is

nonblocking so long as any trunks in the overloaded group are carrying routine traffic. The fractional call cancellation approach will not help in severe overload cases where all calls on the overloaded group are above routine, since fractional cancellations affect only routine traffic; in such cases, the preemption system simply gives preferential treatment to higher precedence levels at the expense of the lower ones.

The mixing of "preferred" and "final" routed traffic on a trunk helps the DSN-Pacific routing scheme to be robust with respect to network damage, but it makes the trunk group controls CANT and CANF relatively ineffective in dealing with overloads, even for routine traffic. A more effective control would be one that allowed the cancellation of traffic that overflowed its routing chain, rather than having been offered to or overflowed from a particular trunk group. Such a control is certainly possible, in that the switch can readily differentiate between these two categories of traffic, but no such control is currently implemented in DMS switches. Among our thoughts for further work in simulation studies are the implementation and testing of such a control, as well as trying the introduction of dummy trunk groups with zero capacity into the routing tables to provide a way to place the existing CANT/CANF controls more effectively.

3.6. CONCLUSIONS

It is attractive to imagine encapsulating the results of our simulation studies in a crisp set of prescriptions coupling clearly recognizable patterns with specific problem types and precise recipes for control actions to correct them, e.g., CB and SK controls for a switch outage. The evidence is mounting, however, that clear-cut problem-response situations are in the minority. Recognition of other problem categories we have examined can be reasonably straightforward, but relieving them through NM action often requires complex combinations of control actions that need to be worked out taking into consideration much detailed information about the state of the network and estimates of the current and expected near-future traffic patterns. From this situation, we conclude that there is far greater challenge for NMES in developing good control action plans than in achieving effective problem recognition.

It is clear from our experiments that some control actions, e.g., reroutes, are needed to achieve the primary NM goal of providing nonblocking service for precedence traffic. Others, such as the use of CANT and CANF to deal with overloaded trunks, offer only modest performance improvements for routine traffic and may not be cost-effective with respect to the DSN mission. A possible conclusion to draw here is that the latter kind of procedures should not be considered for DSN. We are not yet prepared to support such a conclusion wholeheartedly, since we have two concerns in this area. One concern is that, if the cost of implementing a control action is sufficiently low, its use may be warranted even though its benefits are small and limited to routine traffic. Routine traffic constitutes the bulk of the network load under normal peacetime conditions, and budget considerations are likely to cause the network to always be overloaded by such traffic at busy times of day. If the cost of control actions that can give even a small improvement is not high, they should be considered. If nothing more, they can provide useful exercise for human controllers and expert systems in the field.

Our second concern in the area of overloaded trunks is that, while the currently available control actions benefit only routine traffic and precedence traffic is not sufficiently heavy to cause overloading

under normal conditions, in the event of severe damage it is quite possible for trunk resources to be overloaded such that precedence traffic is no longer nonblocking. The control actions that offer some benefit for routine traffic overloads do not help at all for precedence traffic overloads because the relevant controls cannot act to cancel such traffic. However, it does not seem that it would be difficult to either extend the controls to act on precedence traffic or to provide a special mode in which they could be effective for such traffic. In that case, experience with the control actions for routine traffic would be useful training for handling the worst case of precedence traffic overloading. Consequently, we do not advocate abandoning the study of overloaded trunk control actions even though they benefit only routine traffic under the presently defined controls.

4. DCOSS CONTROLLER TRAINING SYSTEM

Task III of the FY88 SOW required demonstration of the feasibility of creating a DCOSS controller training device. This work was completed early in the year, and the DCA asked Lincoln to begin actually implementing such a device. This section combines the reporting of the feasibility demonstration with a description of the status of the DCOSS Operator Trainer development.

4.1. BACKGROUND

In conjunction with the development of DCOSS, DCEC has long recognized the need for DCOSS control personnel to quickly acquire DSN network management skills and experience. The methods of training and apprenticeship normally employed for technically demanding operations functions cannot be used here, since there is no preexisting cadre of skilled professionals to provide the training; indeed, DCOSS itself does not yet exist. When it does get implemented, DCOSS will give the operators far more power to improve performance and correct problems in the DSN than was ever available to AUTOVON network managers; unfortunately, they will also have the power to degrade network performance or even cause disaster through incorrect use of their control capabilities. Consequently, it would be hazardous to expect inexperienced DCOSS controllers to acquire on-the-job training experimentally on the real DSN. Moreover, even if this risky approach were taken, it might be quite some time before the controllers see examples of the more unusual and severe network problem conditions they must be able to overcome.

DCEC also recognized that CCSIM has an excellent potential as the core of a DCOSS controller training system. CCSIM already produces a flow of statistics reports on the simulated traffic flowing through each of the switching nodes in the simulated network, and already has the capability for damage and overload conditions to be imposed and for control actions to be accepted; in short, it is much like the world as seen by a DCOSS network manager. There appeared to be two basically different ways in which CCSIM could be used to support training. One would be to add the capabilities of a DCOSS workstation to the IDSIM configuration and produce a stand-alone training facility. The other would be to modify and extend CCSIM to make it look like the network and connect it to a real DCOSS workstation. In either case, it would be necessary to provide mechanisms for the Training Supervisor to use in setting up problems and evaluating performance for the trainees as well as guidance for a Training Supervisor on how to create realistic scenarios.

We began our feasibility study by investigating the capabilities and requirements of the DCOSS design that was then being vigorously pursued by the Air Force Sacramento Air Logistics Center (SM-ALC). Several visits were made to SM-ALC and to their DCOSS operator workstation contractor, RD Labs, Inc. It was immediately apparent that the complexity of the DCOSS workstation was sufficiently great that incorporating its functionality into IDSIM would be far too costly to justify. We therefore focused our attention on the second approach, i.e., developing a suitable interface between CCSIM and a real DCOSS workstation. The issues that had to be addressed included identification of the best point for the Trainer to tie into DCOSS, development of an operations philosophy for the

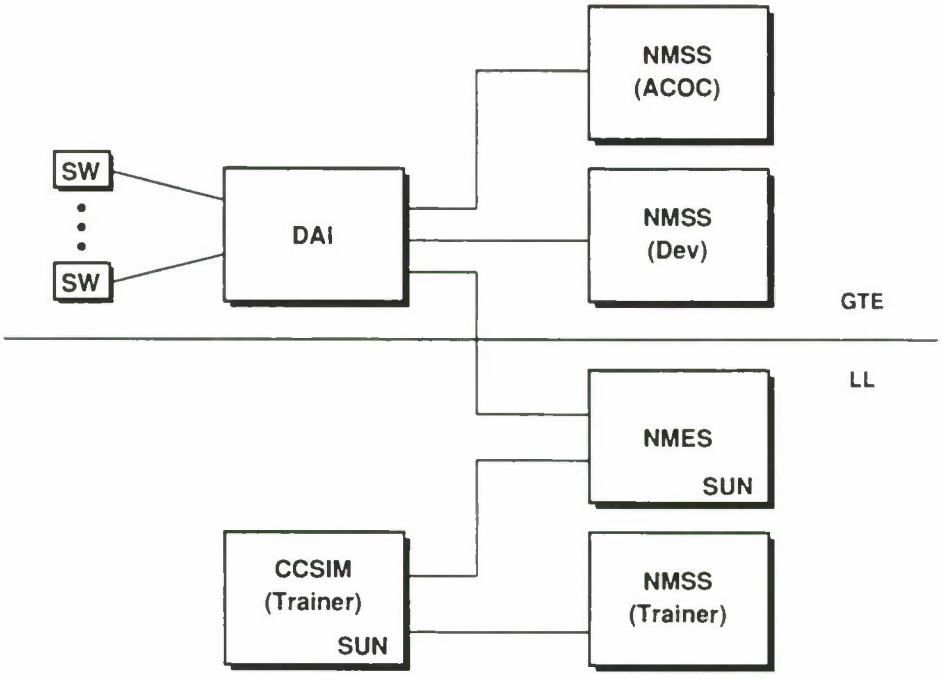
Trainer, and preparation of a detailed specification of the information that would flow in both directions between DCOSS and the Trainer. These issues were resolved and the engineering design was developed under Task III of the SOW. We concluded that the Trainer concept was feasible, and DCEC generated a new SOW and sponsorship for a realization of the Trainer with work starting in FY88 and extending into FY89.

Before our work had progressed beyond the design stage, a decision was made by the DCA to discontinue the DCOSS development effort with SM-ALC. Instead, a new procurement is to be undertaken for a commercially produced DCOSS. In the meantime, DSN network management support is to make use of an existing PC-based Network Management Support System (NMSS) that has been created by GTE contract engineers at DCA-Europe (DCA-Eur). We were directed to aim at achieving the same DCOSS operator Trainer functionality intended earlier, but interfaced with the NMSS. Accordingly, we set aside the designs that would have operated with SM-ALC, and proceeded to develop the NMSS version. The following section describes the current status of its design and implementation.

4.2. TRAINER SYSTEM DESIGN

Figure 4-1 illustrates the existing NMSS, in the region above the horizontal line. At the left are the remote DMS-100 switches, of which there are six at present (five in the UK and one in Spain), with more expected to be connected to NMSS by the time of the delivery of the Trainer to DCA-Eur. The MAP (Maintenance and Administration Position) ports of these switches are connected via modems and telephone lines to a DAI (Data Acquisition Interface) at DCA-Eur. Not shown at the left of the DAI is a "smart switch" giving the DAI access to the modems at the DCA-Eur end of the connections. The DAI itself is an IBM PC/AT running a GTE-produced program that polls each remote DMS-100 once each 15 min for a formatted summary of call processing and trunk utilization statistics. The DAI converts these reports to a standard comma-separated form that is compatible with commercial spreadsheet and database practices, and outputs the results in parallel to the two identical NMSS workstations at the upper right in Figure 4-1. One of these workstations is used operationally in the ACOC, and the other is used by the GTE engineers for follow-on development of the system. Each workstation is an IBM PC/AT which displays the switch statistics data in both text and graphics form for analysis by an operator, and accepts the operator's network management control commands for transmission to the switches via the DAI.

The bottom half of Figure 4-I shows the two subsystems to be delivered to DCA-Eur and demonstrated at the end of FY89. The planned Network Management Expert System (NMES) demonstration is described in Section 5 of this report. The Trainer consists of a copy of the NMSS workstation (shown at the lower right), driven by the Call-by-Call Simulator incorporating the Trainer Interface software system discussed above. The CCSIM and Trainer Interface will run in a SUN 3/260 workstation, and the NMSS workstation PC will be identical to those at DCA-Eur. (Note that a change from the IBM PC/AT to the IBM PS/2 is planned at DCA-Eur, and that the Trainer will have to accommodate to this change.) Not shown in the figure is the Training Supervisor's console, which will resemble the present CCSIM console but will have streamlined setup and operations features tailored to the needs and skills of ACOC personnel.



115282-4

Figure 4-1. Proposed NMES demo/Trainer configuration.

In using the system, the Training Supervisor will preload CCSIM with a file representing the network configuration and traffic conditions as seen from the ACOC, turn CCSIM on and let it run (possibly with simulated time progressing faster than wall clock time), and inject damage or overload conditions at times and locations of his choosing. Documentation to be provided with the Trainer will give guidance for the Supervisor on the introduction of damage scenarios. The trainee will be required to diagnose the problem and apply control actions via the workstation, and CCSIM will respond to the controls. At the termination of each exercise, logs of the activity will be available for review and critique by the Training Supervisor. It will be easy for the Supervisor to present the trainee with exactly the same scenario as many times as desired, to illustrate the effects of various control strategies.

A sample of archived NMSS data files, showing one report from each of three DMS-100 switches, was obtained from DCA-Eur during a visit in September 1988. These data are in the comma-separated format of the DAI output and will be useful for checking the format of the data that the CCSIM/Trainer Interface system must produce in order to drive the NMSS workstation correctly. We expect the form and content of the comma-separated data to change somewhat with the conversion of NMSS to PS/2 computers, but we do not expect these changes to significantly impact our interfacing effort.

In addition to building the Trainer Interface to handle the reformatting issues associated with communications between CCSIM and NMSS, we are making a number of changes and additions to CCSIM itself. We are adding a capability to specify an arbitrary number of trunk groups between pairs of switches and to associate trunk group numbers and Common Language Link Identifier (CLLI) names with these groups. The groups can have new properties such as data quality as opposed to voice quality and can be one-way-only groups, a feature not currently offered by CCSIM. Since some of the switches

to be monitored by NMSS will be end offices, we expect to change CCSIM to handle routing in a slightly different way so that routing tables will not become excessively large as the number of end offices grows. We will also need to do code (telephone number) translation because a switch can be the destination for many end-office codes. CB controls in the real network are invoked in terms of the actual codes to be blocked. In the current CCSIM they are invoked using switch names.

Another feature required for the Trainer is an ability to simulate switch congestion and to generate the associated reports for CPU load, call processing block usage, Receiver Attachment Delay Recording (RADR), and Dial Tone Speed Recording (DTSR). Since the DMS switches being monitored should not go into congestion with the levels of inter-switch traffic that the trunking can support, we deduce that high levels of intra-switch traffic would be needed to produce congestion effects. CCSIM does not currently model intra-switch traffic, and we do not propose to add such a capability because it would significantly slow simulations with little benefit. Instead, we propose to model the symptoms of congestion by generating the relevant reports from average levels of inter-switch traffic multiplied by a factor that the Training Supervisor can adjust on a switch-to-switch basis with a random component added in for verisimilitude. With this mechanism, the Supervisor should be able to produce realistic-looking switch-congestion scenarios.

The principal NM action in dealing with switch congestion is to turn on Line Load Control (LLC) either manually or automatically as a function of CPU load. We plan to extend CCSIM to deal with this control and have its effects appropriately reflected in the relevant reports. We will need to acquire knowledge from the field to correctly model the dynamics of the interaction.

5. NMES FIELD DEMONSTRATION PLANS

During FY88, DCEC decided that a field test and evaluation of NMES should be planned for and carried out at DCA-Eur toward the end of FY89. This decision has considerably influenced our FY88 activity, in terms of both making NMES implementation choices and planning our FY89 work toward the demonstration objectives. In fact, two Lincoln staff spent a week at DCA-Eur in September co-ordinating needs and plans with the site personnel, for both the NMES demo and the DCOSS Operator Trainer. The purpose of this section is to summarize our current views on the NMES demonstration preparations.

5.1. BACKGROUND

As noted in Section 4.2 above, six Northern Telecom DMS-100 DSN switches in the European theatre are presently connected to the Network Management Support System (NMSS) at the ACOC via modems and long-distance telephone circuits, with more to come. The NMSS is a PC-based system for collecting statistics reports from the switches and displaying them in text and graphics form for the ACOC operators. As shown above the horizontal line in Figure 4-1, the Data Acquisition Interface (DAI) PC collects the raw data from the telephone switches and (after putting the data into condensed comma-separated form) passes it on to the two identical NMSS workstation PCs for display. The operator at an NMSS workstation also has the capability to enter control commands for transmission to the switches via the DAI, and this is the expected mechanism by which ACOC DCOSS operations personnel will take action to overcome DSN traffic, routing, and outage problems with respect to the switches monitored by NMSS. Note that this mechanism is not yet fully available: for the time being, ACOC operators must request that local switch personnel implement the desired controls, since the base commanders responsible for the switches have not yet agreed to permit direct control of their switches by an outside agency.

5.2. OBJECTIVES

The objectives of the NMES field test and evaluation are to place NMES in an operational environment, feed it the same information that human controllers using NMSS would have, and observe its performance. In the test situation, NMES would not actually invoke any controls, but would simply report what action it would propose for such problems as it could recognize. As a minimum, the NMES should be able to identify failure of any of the monitored switches and their associated trunk groups, and to recommend network management control actions as appropriate to achieve the best performance possible under such degraded conditions. In additionto the switch statistics provided by the DAI to NMSS, NMES will be given collateral information about the network that reaches the ACOC by other means, such as verbal reports by telephone.

One of the issues that must be addressed in conducting the NMES field demonstration is that the DMS switches are reliable, modern equipment, not prone to failure. As a result, there may be little

anomalous activity in the live switch reports for NMES to detect and identify, and the demonstration could be very dull. Referring to Figure 4-1, we see that more variety can be added to the demonstration by connecting the DCOSS Operator Trainer to NMES for a period of time, and requiring NMES to correctly deal with congestion and damage scenarios introduced into the Trainer.

5.3. APPROACH

The NMES implementation presently at Lincoln Laboratory has dealt with a stream of switch statistics reports provided by CCSIM in a format convenient for processing in IDSIM. Each switch in the simulated network reports once in each 5-min interval. In order to perform the desired field testing at DCA-Eur, the NMES design must be modified to take account of the current NMSS environment at DCA-Eur. To this end, we have studied the NMSS architecture, discussing it with GTE and DCA-Eur personnel, and have concluded that the best interface choice is to drive NMES with the same DAI output data that feed the NMSS workstation. Figure 4-1 shows this interface connection from the DAI to the NMSS. As noted in Section 4, we have obtained files of archived NMSS data from DCA-Eur that will, in principle, give us all the guidance necessary to correctly modify the NMES front end to receive and parse data from the DAI. In addition, the NMES modifications must provide a way for collateral information about the network to be entered manually. Similarly, NMES must be informed of what control actions, if any, have been taken by the network managers, since it cannot assume that its recommendations have been accepted.

Another issue we will have to face is that it takes time to apply NM controls in the real world, and, in fact, some take longer than others. For example, the Network Management NM/NMSS Handbook indicates that applying CB in response to a switch outage could take as much as 3 or 4 h. The Handbook suggests that, in order to prevent the spread of switch congestion before the CB commands take effect, 100-percent CANT should be applied at all switches having direct trunk connection to the damaged node. Similarly, it may be necessary for us to determine intermediate actions for NMES to take while we are waiting for controls to take effect.

The ACOC DCOSS Operator Trainer system described in Section 4 offers a convenient aid in developing and testing the field-test version of NMES at Lincoln Laboratory. As shown in the central portion of Figure 4-1, the NMES should be able to transparently receive and act upon either real DAI data or simulated data from the Trainer, since both kinds of data follow the same specifications as to format and contents. Moreover, the Trainer offers far richer opportunities to test NMES under widely varying overload, failure, and disaster scenarios. The real switches are likely to be operating normally most of the time, and disaster conditions are likely to occur very seldom, so one could finish up waiting a very long time for corroboration that NMES can properly handle an interesting variety of real problems. We expect that a successful field demo and evaluation will include NMES testing on both real and simulated switch reports.

6. APPLICATION OF EXPERT SYSTEM TECHNIQUES TO DDN

Task IV of the FY88 Statement of Work provides for analysis and review of future requirements of the Defense Data Network (DDN) for monitoring and control, and for identification of expert system techniques and technology that could be applied to improve the effectiveness of DDN network management. This work was carried out primarily in the first half of the year, and an interim report was presented at a briefing at DCEC on 10 March 1988. This section describes the work and summarizes the recommendations we developed. Basically, of the seven kinds of activity at DDN Monitoring Centers we found two (resource management and crisis management) requiring complex situation analysis and choice of corrective action plans in real time, such that expert system technology could be applied (albeit at considerable cost) to enhance operator performance. In both cases, augmentation of the expert system with an operator training simulator would be very useful. As explained below, the other five kinds of activity could benefit considerably by adding more conventional automation and database management systems, but there is no clear justification for expert system application.

Four attributes distinguish a potentially fruitful application area for an expert system, and all four must be present if the cost and labor of the expert system implementation are to be justifiable:

(1) Complex, expertise-intensive tasks requiring long experience, extensive training, or both for humans to perform them effectively;

(2) A workload consisting primarily of instances from a modest-sized repertoire of such tasks;

(3) A considerable degree of pressure, in that the tasks must be performed quickly, and many tasks may be in progress at once; and

(4) The possibility of extended, detailed knowledge engineering interaction with sources of expert knowledge in the application area.

Three considerations entered into our study: (a) our experiences with modern expert system shells in development of expert systems for military communications network management and control; (b) an assessment of each DDN MC function in terms of attributes (1), (2), and (3) above; and (c) an assessment of the trade-offs between the cost and labor of knowledge engineering and expert system implementation in each task area, vs the manpower savings and efficiency improvements possible if that expert system were fully available and perfectly implemented.

The expert system shells themselves can lead to considerable efficiency in implementation, in that many overhead items (such as rule statement protocols, memory management, and inference mechanism implementation) are already done. Many such shells are available, and each has its advantages and disadvantages. It is not the purpose of this report to compare shells, or to analyze the current state of the art in expert system development; suitable treatments of these topics are found in recent conference proceedings of the AAAI (American Association for Artificial Intelligence), for example, and in Proceedings of the IJCAI (International Joint Conferences on Artificial Intelligence). For the two DDN MC functional areas cited here, we believe that any of the modern shells (ART, KEE, CLIPS, ...) could be

used successfully, and the choice could be guided by considerations such as costs and the capabilities of the available programming staff. In any case, however sophisticated or expensive they may be, shells and tools cannot significantly diminish the knowledge engineering work of iteratively analyzing the knowledge in a domain of expertise and reducing it to a (generally quite large) set of "if-then" relationships or other formalisms that can fit within the machinery of the shell in use. Our study primarily asks whether the DDN MC functions fit the expert system paradigm, and, if so, whether the projected benefits outweigh the labor of knowledge engineering.

6.1. INFORMATION SOURCES

A useful source of background information was a report loaned to us by DCEC/R640 entitled "DDN Monitoring Center Functional Description," BBN Computer Company Report 6138, which was written for DCEC in February 1986. Building upon this basis, our study efforts entailed visits, tours, and discussions with the following organizations and individuals, coupled with analysis of the information thus obtained as well as study of relevant documentation, in the light of our DCEC-sponsored DSN Network Management Expert System development experience to date.

(1) Mr. Ramon C. Butler, Chief, DCA Switched Networks Branch, Arlington, VA (1 September 1987)

(2) DCA Operations Center, Arlington, VA (5 November 1987)

(3) Telenet International Network Control Center, Reston, VA (1 December 1987)

(4) SRI Packet Radio Network Management Project, Menlo Park, CA (5 January 1988)

(5) BBN Automated Network Management Project, Cambridge, MA (15 January 1988)

(6) DCEC/R640 staff members, Reston, VA (21 January 1988)

(7) DARPA Internet (including ARPANET) Network Operations Center, BBN, Cambridge, MA (25 February 1988)

(8) AT&T Corporate Voice and Data Network Management Center, Orlando, FL (10 May 1988).

Sources (1) and (2) provided us with verbal descriptions, tours, and demonstrations of the current practices of DDN Network Management, as well as discussions of perceived future needs (summarized in Section 6.2 below). Sources (3), (4), (5), (7), and (8) are commercial data communications organizations which are motivated by profit potential to apply any useful technology advances to improve their performance and efficiency. All sources allowed us to tour their operations, and spoke freely about their evaluations and applications of expert systems technology, which we found quite illuminating as described below. Source (6) was an interim review with DCEC personnel which gave us useful insights about further analyses to pursue in our study.

6.2. THE PROBLEM DOMAIN

The DDN is a wide-area military packet data network based on the X.25 protocol. It provides terminal-to-host and host-to-host connectivity for DoD-authorized users. It consists of five component networks, as follows. The ARPANET (although not actually a part of the DDN) is operated by DCA for DARPA. The MILNET is an ARPANET look-alike within the DDN, and it provides unclassified general service. The DISNET (Defense Integrated Secure Network) is a system-high Secret general service net. The WINCS (WWMCCS Inter-Computer Network, Communications Segment) is a Top Secret net dedicated to WWMCCS support. The SCINET (Sensitive Compartmented Information Network) serves a special community. These component networks all employ distributed packet switches performing many traffic, routing, and error-recovery functions. The operation of these networks at multiple security levels is based upon physical separation at present, and will be based upon the BLACKER end-to-end encryption system after its expected deployment in the 1989 time frame.

Centralized network management of the DDN is done via the DDN Monitoring Center System (DMCS) at present. Numerous sophisticated data-gathering and presentation tools are used in this effort, such as the Bolt Beranek and Newman (BBN) Network Utility (NU) system; however, these tools depend entirely upon human operators to assess and evaluate the data, to compare it with expected operation, and to take action to return unsatisfactory operation as nearly to normal as possible. Currently, nine Monitoring Centers (MCs) are in operation, and Sections 6.2.1 through 6.2.7 below describe the seven generic categories of management activity carried out at the MCs. Three are at the DCAOC in Washington, for MILNET, DISNET, and WINCS. One is at the Pacific Theatre ACOC (Wheeler AFB, Hawaii), and one is at the European ACOC (Patch Barracks, Germany). There are WINCS backup MCs at Ft. Ritchie and DCEC, and the SCINET MC is on the premises of the special community. The ARPANET MC is integral with the DARPA Internet NOC at BBN in Cambridge, Massachusetts, and it also handles MILNET installation management.

There are current needs for DMCS monitoring center upgrade and expansion, driven by such factors as the rapid growth of MILNET and the addition of user service center functions and mutual backup functions to MCs. Whether such upgrades could profitably include expert system technology, or should focus on more conventional automation techniques, is discussed below.

Sections 6.2.1 through 6.2.7 provide background descriptions of each DDN MC function category. Section 6.3 reports the observations of commercial organizations about expert system applicability for similar functions in their operations. Section 6.4 summarizes our recommendations about the potential value of expert system application in each DDN MC function area.

6.2.1. Management Control

The goal of this MC function is to maintain and support DCA policies and procedures for the DDN. The activities involved include task prioritization, staff scheduling, and activity supervision in relation to all the other MC functions. Management control functions are basically longer-range in character, and do not require decisions or actions under real-time pressure.

6.2.2. Problem Management

These activities include diagnosis of faults in the networks and at subscriber interfaces, and the initiation of actions to resolve them. In addressing network faults, human operators at consoles in the various MCs run network utility programs which allow them to: poll remote packet switching nodes for status information; view alphanumeric summary displays of the resulting data, for their reference in identifying the probable cause of trouble; and transmit commands to remote sites to reload and restart their packet switch control programs, for example, or to change parameters in running programs. The majority of typical network problems are solved in this way, and the remainder require the dispatching of a computer technician to the site, provided with symptom descriptions from the MC operator. Provisions are made for programmed escalation and notification of outages as time elapses during network fault diagnosis and recovery efforts.

Subscriber interface problems vary widely, ranging from minor to exotic, because there are many varieties of subscriber equipment and because the site personnel are likely to have little or no knowledge of the network or its operation. The complaints can sometimes be resolved through telephone negotiation and instruction to the subscriber who has the problem, but sometimes problems persist until a knowledgeable network resource person can be dispatched to the site.

6.2.3. Configuration Management

The goals of this function are to maintain a database for desired and actual network configurations, and to verify compatibility and authorization for changes. Like Management Control, this function is longer-range: actions are planned and carried out on a time scale of weeks or months, rather than minutes or hours. The function is basically administrative, and depends heavily upon keeping the database accurate and complete.

6.2.4. Resource Management

This function has many parallels with the initial engineering of a data network, in which packet switch and transmission resources are sized and procured to deliver the desired performance under the anticipated loads. The primary difference is that resource management must be done in real time if it is to have any value: the sizes and capacities of resources cannot be increased on such a short time scale, but one may attempt to reallocate the available resources to accommodate loads or conditions that are found to differ markedly from those for which the network was engineered. Such differences could include data traffic overloads or distortions, or loss of some resources through failure or damage. Although implementation of this function in the DDN has not yet actually begun, its intended goal is to recognize degraded network performance and to better match the network capabilities to the current traffic by shedding load or reallocating trunking. This will involve the gathering of traffic and performance data, analysis of the data to recognize problems, and devising of control strategies to achieve the best possible performance with the remaining assets.

6.2.5. Security Management

These activities include rekeying trunk encryption devices and enforcing TAC access restrictions, and may include additional roles in the post-BLACKER environment. The present complex manual procedures required for key distribution, rekeying of cryptos, and re-establishment of crypto synchronization can produce heavy workloads and can cause excessive trunk outages. Modernization of equipment and introduction of auto rekeying can substantially solve these problems.

6.2.6. Crisis Management

The current capability in this area is limited to providing alternative MCs as backup. The goal for this function is to extend the resource management capabilities described above to provide for network reconstitution after damage events. This can be a very complex process, effectively a real-time version of the current engineering labor-intensive, offline procedures for creating the initial design of a network and its routing structure. An especially important problem in reconstitution of a fragmented network is obtaining and validating accurate database information about the structure of each fragment, then insuring consistency among all the elements of the reconstituted network.

6.2.7. Data Management and Reporting

These functions include support of the database needs of the other MC activities, preparation of both routine and ad hoc reports, and maintenance of archives of network performance. This kind of activity requires little skill, in general, and can typically be done by junior people with moderate training and supervision.

6.3. OBSERVATIONS BY COMMERCIAL ORGANIZATIONS

In general, we found that the commercial companies visited were well aware of expert system capabilities and had given serious thought to whether they could obtain worthwhile advantages by this means. Their conclusions are summarized as follows.

6.3.1. Telenet NCC

Mr. Robert Lipp, an upper manager at the Telenet International Network Control Center in Reston, Virginia hosted a visit by Lincoln and DCA personnel (G. Coviello and C. DiFiore). The NCC supervisor described their primary problem domain, which is their equivalent of the DDN MC Problem Management function described above. (The other DDN MC functions which are relevant to Telenet — Subsections 6.2.1, 6.2.3, and 6.2.7 — are handled by their administrative staff, not by the NCC.) Telenet has 23,000 mi of optical fiber in CONUS, plus other assets, and their network access is exclusively X.25. They have satellite links to international customers, with packet switches at the earth stations. Their International Network Control Center in Reston employs a Plessey network monitoring software system running in a Prime 750 computer. They use three operators per shift, during normal working hours only. These operators are typically former military Tech Controllers who have received one year of additional training plus ongoing on-the-job training.

Trouble reports are taken by customer service representatives; in fact, many problems are solved by these people by instructing customers to make simple checks and verifications. Of the faults that get referred to NCC shift personnel, the hardware problems tend to be easy: the switches have remotely controlled test equipment, and fault isolation is generally straightforward. Although Telenet has not found it worthwhile to do so, it would be possible to create a software system that could pre-screen customer service calls for such hardware problems — not an expert system, but a simple computerized checklist. Development of a generalized expert system for customer service interactions would be very difficult, because service calls are typically either the simple type just noted, or complex situations that require common sense and creative analysis. Protocol problems are the worst, typically resulting from varying interpretations of standards by various equipment manufacturers. Protocol problem symptoms are complex, tests are difficult to perform and to analyze, and each case tends to be unique. Telenet people have concluded that expert systems do not have a good application in their NCC at present: the problems tend to be either too simple to need AI, or so complex and unique that the development of even a moderately comprehensive knowledge base for an expert system would be a prohibitively formidable and expensive task.

6.3.2. SRI Packet Radio NM Project

Dr. Earl Craighill of SRI International discussed their network management issues at length with Lincoln personnel. SRI has been involved for some years with the development of networking technology for low-cost packet radios, which use packet switching techniques with extensions that permit frequent changes of network topology as mobile packet radios join, leave, or change their connectivity. There are strong parallels between this environment and the resource management/crisis management aspects of DDN operation, where it is desired that the DDN continue to operate effectively in the face of network changes due to node or transmission system damage or restoral. SRI has conducted a number of packet radio field demonstrations at Army sites, and their network management problems have tended to be unique and very complex; hence, they have not yet made significant progress in developing an organized base of problem-solving knowledge. They believe that expert system technology is clearly applicable here, but that it would be a difficult undertaking.

6.3.3. BBN Laboratories Automated Network Management (ANM) Project

Mr. Bruce Laird of BBN Laboratories (a separate corporate entity from BBN Computer Company, which operates the DARPA Internet NOC described below) heads a project funded by DARPA and the Army Communications Electronics Command (Ft. Monmouth) to develop automated network management technology for the 300-element Internet and for Army tactical networks such as mobile packet radio nets. Mr. Laird is pursuing three areas of effort: distributed architectures, human interfaces, and expert systems. Until now, the ANM capabilities contemplated are limited to monitoring, not actually controlling.

BBN's thoughts on Problem Management are discussed in the following section. Mr. Laird's group plans to address a perceived need for two kinds of expert systems: an End-to-End Connectivity Expert and a Configuration Management Expert. The former concept is applicable to the DDN Resource

Management and Crisis Management functions, in the sense of maintaining expert knowledge of possible alternate routes for various contingencies. The BBN Configuration Management concept parallels that of the DDN and, in fact, it was noted above that BBN manages MILNET installations. In this context, it may be noted that BBN Computer Company is already using Design-Net (an AI-like tool they have developed) to help their sales representatives with the complex tasks of configuring BBN host computers being sold to customers; however, this function has no direct parallel in the DDN MC. For the basically administrative functions of Configuration Management in the DDN MC, BBN feels that automated tools and database management software can help greatly, but expert systems have no good application.

6.3.4. DARPA Internet Network Operations Center (NOC), BBN, Cambridge, Massachusetts

Lincoln personnel were given an extensive tour and description of the NOC by the operations supervisor. This center manages the ARPAnet and a number of other university and research networks interfaced with it, all of which together constitute the Internet. In a sense, the NOC is the archetype of the DDN Monitoring Center Problem Management functionality; in fact, MILNET monitoring in DDN MCs is done by BBN employees using the same tools. Shift personnel sit at the consoles of C70 computers running NU (Network Utility) software, which is a collection of programs developed over the years for efficient performance of the functions described in Section 6.2.2 above. The NOC personnel feel that these tools allow an experienced human operator to perform efficiently in solving problems, and that, in fact, the breadth and variety of Internet network and subscriber interface problems are such that it would be impractical to build an expert system with a wide-enough range of knowledge to be useful.

6.3.5. AT&T Corporate Network Management Center

AT&T has built a large modern facility in Orlando, Florida into which they have consolidated the management and control of a number of corporate voice and data networks. These functions include Tech Control of transmission facilities (accessing modern remotely controllable communications and test equipment), and indeed the primary purpose of the visit was to study their Tech Control implementations in pursuit of Lincoln Laboratory work in that area. The AT&T facility also manages several data networks in the DDN MC sense and, although many details of the AT&T networks were different from those discussed above, there was a clear similarity in the kinds of observations made about automation and expert systems applicability. They felt that expert systems were an interesting notion for the future, but that at present it would be uneconomical to try to build them.

6.4. SUMMARY OF DDN MONITORING CENTER EXPERT SYSTEM ISSUES

Briefly stated, our studies indicate that improvements in efficiency and accuracy can be realized by exploiting conventional kinds of automation in five of the seven categories of DDN MC functions, while expert systems combined with training simulators could be of significant help in the other two areas, though they could be costly.

Management Control (Subsection 6.2.1) lacks Attribute 3 (time pressure) as described in Section 6.2, hence the cost of developing an expert system for this function is hard to justify. Off-the-shelf scheduling and word-processing software tools can help greatly in reducing the workload by maintaining lists and schedules and producing required paperwork for Management Control. Attribute 3 is also missing in Configuration Management and Data Management and Reporting (Subsections 6.2.3 and 6.2.7, respectively); moreover, Attribute 1 (complexity) is only weakly satisfied. Scheduling and word-processing tools could help reduce operator workload, and a modern database management system would be highly desirable for accuracy and convenience in maintaining records of the system configuration and archiving performance data. Since humans are readily trained to use these tools effectively, and can then perform the required functions satisfactorily, there is no motivation to try to build expert systems for further improvement.

In Problem Management (Subsection 6.2.2) and Security Management (Subsection 6.2.5), we have remarked that replacement of old manual equipment with modern automated or remotely controllable products can greatly improve personnel performance and reduce workload. We have seen how remotely controlled test equipment at network nodes makes many of the network fault isolation events very straightforward for the network managers in the commercial organizations visited, and similar goals should be sought by the military. This would also reduce the number of service calls by technicians to remote sites. Similarly, subscriber interface problem solving would be helped greatly by such upgrades as replacement of customer-premises modems with modern "smart" modems capable of remotely commanded loopback. Much crypto equipment in current use requires slow manual loading of keys and tricky, time-consuming manual end-to-end resynchronization, during all of which the trunks affected are out of service. Modernization of this crypto equipment would lead to more workload and efficiency improvements.

After these straightforward improvements were done, the remaining tasks in Problem Management and Security Management would be the widely varying special cases such as the protocol problems noted above, causing these functions to fail to have Attribute 2 (modest-sized repertoire of types of problems). This implies that a great deal of work would be necessary to implement enough varieties of special cases in an expert system to be of significant value. Moreover, these problem areas tend to require a degree of common-sense capability that is beyond the state of the art in AI. Consequently, the motivation for expert system development in these areas is not present.

Given that a core of good customer-services representatives must be maintained to handle these special tasks, one should not consider expert system development to handle the more straightforward tasks in Problem Management and Resource Management. The person or persons on duty can easily use the automated, remotely controllable modern equipment recommended above.

The remaining two DDN MC functions, namely Resource Management and Crisis Management (Subsections 6.2.4 and 6.2.6), both possess all four of the attributes in Section 6.2. They are distinguished by complex situation assessment and corrective action planning requirements, by a moderate number of types of problems, by time and workload pressures, and by shortages of experts who know how to do the required functions. Also, the stress conditions that will require these functions seldom or never occur in advance to permit practice or training in dealing with them, yet must be countered with

fast and precise action when they do occur. In all these respects, these two functions are similar to the DSN Network Management environment that Lincoln is addressing under DCEC sponsorship. Given the required motivation and support, we suggest that a similar approach could be taken in these two areas to that in progress for the DSN: construct a training simulator that creates representations of the stress conditions that must be managed, and is able to accept the corrective actions that the trainee applies. Concurrently, accumulate a knowledge base of experimentally derived expert knowledge on detecting and correcting problems, and embed the knowledge in an expert system. These two processes would no doubt entail considerable cost and effort, and would lead to capability enhancements rather than operating cost reductions; to the extent that these capability enhancements are an important gain for the Government, the costs might be justifiable.

APPENDIX A
CCSIM NETWORK MANAGEMENT CONTROLS

Three types of controls implemented in CCSIM are listed below. Pre-route controls are applied at a switch before an attempt is made to route the call out of the switch. Pre-hunt controls are applied at a switch before an attempt is made to find a free trunk in a particular trunk group. Post-hunt controls are applied at a switch after the call has overflowed a trunk group. Within each group, the controls are listed in the order in which they are applied. The term DMS in the right-hand column indicates that the control is implemented in accordance with Northern Telecom Practices for DMS switches. The term 490L after the DRZ control indicates that this is the directionalize control as implemented in the AUTOVON 490L switches, and has been implemented in CCSIM at the request of DCEC to support certain simulation studies. A blank entry in the third column indicates that this control is implemented according to the generic switch specification. In all cases, the implementation functionality was arrived at by carefully studying the documentation, discussing the design details in depth with knowledgeable DCEC personnel, then carefully testing the CCSIM implementation to verify that it meets the design criteria.

Pre-route Controls:			
1.	Code Block	(CB)	DMS
2.	Call Gap	(GAP)	
3.	Alternate-Route Cancel (B)	(ARC-B)	
Pre-hunt Controls:			
1.	Alternate-Route Cancel (A)	(ARC-A)	
2.	Directional Reservation Equipment	(DRE)	DMS
3.	Directionalization	(DRZ)	490L
4.	Cancel-To (Percent)	(CANT-%)	DMS
5.	Cancel-To (Rate)	(CANT-RATE)	
6.	Skip	(SK)	DMS
Post-hunt Controls:			
1.	Cancel-From	(CANF)	DMS

Calls affected by CB, CANT-%, CANT-RATE, and CANF are connected to announcement messages. There are three such announcements. In CCSIM, Emergency Announcement 1 (EA1) and Emergency Announcement 2 (EA2) mean that the effected call will be counted as failed and will not be retried. No Circuit Available (NCA) announcement allows the call to retry subject to the retry parameters.

In the following descriptions, the term "link" refers to the combination of all trunk groups between a pair of switches.

Pre-route Controls

CB — Code Block

The CB control is put on at a switch and applies to originating calls only. It blocks a specified percentage of the traffic to the destination code from entering the network. When the control is applied at less than 100 percent, only routine calls are affected. At 100 percent, CB blocks all precedence calls. Blocked calls are handled according to their announcement type.

Usage: CB node1 node2 percent ann
node1 is a three letter node name or ALL
node2 is a three letter node name or ALL
percent is the percentage in hundredths(0-100)
ann is the announcment type (NCA, EA1 or EA2)
Example: CB ALL PRL 50 EA2
Cancels 50% of the routine calls from all nodes to PRL.

GAP — Call Gap

The GAP control is put on at a switch and applies to originating calls only. It determines the rate at which traffic to the destination code enters the network. After one attempt to route a call to such a destination, all subsequent calls to that destination are blocked for a period of time designated as the "gap interval."

After the expiration of the gap interval, an attempt is made to route the next call that arrives for that destination. The gap interval is chosen from the interval set 0 (no-control), 0.10, 0.25, 0.50, 1, 2, 5, 10, 15, 30, 60, 120, 300, and 600 s. An infinite interval prohibits all attempts. Traffic blocked by the GAP control is routed to EA2 and will not retry. At levels 0-600, only routine calls are affected. At the infinite interval, all precedence calls are affected.

Usage: GAP node1 node2 index
node1 is a three letter node name or ALL
node2 is a three letter node name
index is a pointer into a table of gap intervals.

Index	Gap Interval (Seconds/Call)	Calls per Minute
1	0	All
2	0.1	600
3	0.2	240
4	0.50	120
5	1	60
6	2	30
7	5	12
8	10	6
9	15	4
10	30	2
11	60	1

Index (Seconds/Call)	Gap Interval Minute	Calls per
12	120	1/2
13	300	1/5
14	600	1/10
15	Infinity	None

Example: GAP ALL FBK 15
Allows no calls from any nodes to route out to FBK.

ARC-B — Alternate-Route Cancellation (Type B)

The ARC-B control is put on at a switch and applies to both tandem calls and originating calls. All calls to the specified final destination are allowed to direct-route out of the switch only. These calls may not alternate-route out of the switch. The values allowed for precedence are R (routine) and AP (all precedences).

Usage: ARC-B node1 node2 precedence
node1 is a three letter node name or ALL
node2 is a three letter node name
precedence is either R, AP or NONE(remove control)
Example: ARC-B SCS FBK R
Routine calls may not alternate-route out of SCS if destined for FBK.

Pre-hunt Controls

ARC-A — Alternate-Route Cancellation (Type A)

The ARC-A control is applied to a link. It causes traffic which would normally alternate-route on that link to be skip-routed. The values allowed for precedence are R (routine) and AP (all precedences). Affected calls skip-route to the next link in the routing table. Traffic that is skip-routed is not counted in the statistics as attempts on the link.

Usage: ARC-A node1 node2 precedence
node1 is a three letter node name or ALL
node2 is a three letter node name
precedence is either R, AP or NONE (remove control)
Example: ARC-A SCS FBK AP
All traffic at SCS that would normally alternate-route to FBK is skip-routed.

DRE — Directional Reservation Equipment

The DRE control is applied to a trunk group defined by its source and destination and by its trunk type (terrestrial or satellite). It gives priority to incoming traffic by reserving a number of idle trunks in the group. When the number of idle trunks is equal to or less than the number of reserved trunks, all

traffic (direct- and alternate-routed) is skip-routed. All-precedence and routine calls are affected by the DRE control. Traffic affected by DRE is not counted in the statistics as attempts on the trunk group.

Usage: DRE node1 node2 type reserve
node1 is a three letter node name
node2 is a three letter node name
type is either LAND or SAT
reserve is the number of trunks reserved
Example: DRE CKS FBK LAND 7
CKS must keep seven trunks to FBK free.

DRZ — Directionalization

The DRZ control is applied to a trunk group defined by its source and destination and by its trunk type (terrestrial or satellite). It places a limit on the number of trunks which may be used for outgoing calls on the trunk group. The level is the number of trunks allowed to carry outgoing calls in the trunk group and ranges from 1 to the maximum capacity of the trunk group. Setting the level to zero removes the control. Traffic (direct- and alternate-routed) which would exceed the limit is skip-routed. All-precedence and routine calls are affected by the DRZ control.

Traffic affected by DRZ is not counted in the statistics as attempts on the trunk group.

Usage: DRZ node1 node2 type level
node1 is a three letter node name
node2 is a three letter node name
type is either LAND or SAT
level is the number of outgoing useable trunks
Example: DRZ CKS FBK LAND 7
CKS may use only seven trunks on its land link to FBK.

CANT-DIRECT-%/CANT-ALTER-% — Cancel-To (Percent)

These controls are applied to a link. They cancel a percentage of the routine traffic offered to the link. CANT-DIRECT-% cancels only direct-routed routine traffic. CANT-ALTER-% cancels only alternate-routed routine traffic. Calls affected by these controls are not counted as attempts on the link.

Usage: CANT-DIRECT-% node1 node2 percent ann
or
CANT-ALTER-% node1 node2 percent ann
node1 is a three letter node name or ALL
node2 is a three letter node name
percent is the percentage in hundredths (0-100)
ann is the cancellation type (NCA, EA1 or EA2)

Example: CANT-DIRECT-% CKS FBK 80 EA1

Cancels 80% of the routine direct-routed calls on the link from CKS to FBK.

CANT-DIRECT-RATE/CANT-ALTER-RATE — Cancel-To (Rate)

These controls are applied to a link. They control the rate at which calls are allowed to access the link. After one attempt is offered to the link, subsequent attempts to that link are blocked for a period of time designated by the "gap interval." After the expiration of the gap interval, the next attempt to the link is permitted. The gap interval is chosen from the interval set 0 (no-control), 0.10, 0.25, 0.50, 1, 2, 5, 10, 15, 30, 60, 120, 300, and 600 s. An infinite interval prohibits all the attempts to the link. CANT-DIRECT-RATE affects only direct-routed traffic. CANT-ALTER-RATE affects only alternate-routed traffic. When the CANT-RATE index is 1-14, only routine calls may be canceled.

When the CANT-RATE index is 15, all-precedence calls are canceled. Calls affected by the CANT-RATE controls are not counted as attempts on the link.

Usage: CANT-DIRECT-RATE node1 node2 index

or

CANT-ALTER-RATE node1 node2 index

node1 is a three letter node name or ALL

node2 is a three letter node name

index is a pointer into a table of gap intervals.

Index	Gap Interval (Seconds/Call)	Calls per Minute
1	0	All
2	0.1	600
3	0.2	240
4	0.50	120
5	1	60
6	2	30
7	5	12
8	10	6
9	15	4
10	30	2
11	60	1
12	120	1/2
13	300	1/5
14	600	1/10
15	Infinity	None

Example: CANT-DIRECT-RATE CKS FBK 10

Allows two direct-routed calls through per minute on the link from CKS to FBK.

SK-DIRECT/SK-ALTER — Skip

The SKIP group control is applied to a link. The SKIP control skip-routes traffic to the next link in the routing table. SK-DIRECT affects only direct-routed calls. SK-ALTER affects only alternate-routed calls. All-precedence calls are affected by the SKIP control. Traffic that is skip-routed is not counted in the statistics as attempts on the link.

Usage: SK-DIRECT node1 node2 percent
or
SK-ALTER node1 node2 percent
node1 is a three letter node name or ALL
node2 is a three letter node name
percent is the percentage in hundredths (0-100)

Example: SK-DIRECT SCS PRL 70
Skip-routes 70 percent of the direct-routed calls on the link from SCS to PRL.

Post-hunt Controls

CANF-AR/CANF-DAR — Cancel-From

The CANF control is applied to the final trunk group in a link. It cancels traffic overflowing from that trunk group and prevents it from continuing to the next link in the routing table. CANF cancels a percentage of overflowing traffic from a trunk group. Canceled calls are handled according to their announcement type.

CANF-DAR affects all alternate-routed calls and the specified percentage of direct-routed calls. CANF-AR affects only the specified percentage of alternate-routed calls. Only routine calls are affected by this control. Calls affected by the CANF control are counted as attempts on the link.

Usage: CANF-AR node1 node2 percent ann
or
CANF-DAR node1 node2 percent ann
node1 is a three letter node name.
node2 is a three letter node name.
percent is an integer from 0 to 100
ann is cancellation type (NCA, EA1 or EA2)

APPENDIX B
RECOMMENDATIONS FOR NM CONTROL ACTIONS IN THE CURRENT DSN

B.1. INTRODUCTION

This Appendix is intended to set forth our recommendations for NM control actions that are applicable for controllers managing the current DSN. At the end of FY87, we produced an issue of our Quarterly Report dated 31 October 1987 that summarized our recommendations as of that date. This document restates and extends those recommendations, taking into account new results obtained in FY88.

We have confined our recommendations here to damage situations and traffic anomalies which we have been able to study through simulation. Certain other situations that require NM action, such as partial loss of switch signaling capability, are not currently represented in our simulator. Although we would have recommendations for many of these situations based on common sense and discussions with experienced network controllers, and some of these are to be found in the body of this report, we have not included them here.

The recommendations are based on the results of simulations of projected DSNs whose configurations and busy-hour traffic estimates have been provided by DCEC. We have carried out simulations using two different Pacific network configurations and one European configuration. Switch behavior and NM controls are based primarily on the DCA Generic Switch Specifications, with some extensions to model Northern Telecom DMS and AUTOVON 490L switches. These simulations do not reproduce the performance of the current DSN in detail and, consequently, we make no claim that any particular measure of performance (such as the overall GOS to be expected in a particular trunk outage situation) would be observed in the real network. However, we do believe that comparisons between alternative NM actions can be treated as valid when other variables are held constant between simulation runs.

The simulations provide both switch reports of the sort available to network managers, and statistics of network performance as seen by its users. We use the switch report outputs to recognize problem situations, and the performance statistics to evaluate alternative NM control actions that might be invoked to deal with the problems. These statistics provide information not available in the real network, such as the number of retries needed to achieve a successful call. Data from the real network give only the total number of attempts in some time period. Similarly, the simulation statistics show which source/destination pairs are having trouble communicating, allowing the fairness of NM actions to be evaluated.

In our evaluations, we have primarily used two performance measures: Grade of Service (GOS), and Call Failure Rate (CFR). GOS is a familiar measure that reflects the probability that a call attempt will be blocked. CFR reflects the probability that an intended call of duration T, between a source switch and a destination switch, will fail to be achieved in spite of retrying the call after experiencing blocking, preemption, or loss due to network damage. We believe that CFR is a better measure of actual network utility for its users than is GOS, since it takes the effects of preemption into account. CFR

values depend on the retry persistence of the users, a quantity that is unknown in the real world, but that can be specified in the simulations. By keeping the retry persistence constant across simulation runs, we can use CFR to make valid comparisons between control options.

If retries are turned off in a simulation, CFR will be worse than GOS because preempted calls as well as blocked calls will count as failures. If retries are turned on, CFR tends to improve as retry persistence increases and will approach an asymptotic value somewhat lower than GOS for most situations. Obviously, if no network path exists for a call, it will fail no matter how many times it is retried. For such calls, both CFR and GOS will be 100 percent. It has been our observation that, for situations where retrying results in success, there is little improvement in CFR between a retry limit of 5 and one of 10. We used 10 in the simulations upon which our recommendations are based.

In making comparisons among control alternatives, we calculate GOS and CFR for traffic that should have some chance of success. For example, traffic to a damaged switch is removed from the calculation since CFR and GOS for that traffic will always be 100 percent, and if such traffic represents a significant fraction of overall traffic, it will dominate the overall CFR and GOS figures and tend to obscure the effects of any control actions.

In our NM experiments, we considered a number of different network damage scenarios and traffic anomalies. The following sections present our specific recommendations for NM actions to be taken for some of the problem situations that were investigated. For many of the damage scenarios, we have identified control actions which we believe to be beneficial, and we make recommendations for such cases. Our investigations of traffic anomalies have not shown many situations in which the benefits of control actions justify the costs and risks involved in implementing them. For such cases, our recommendation is to do nothing. Even in damage cases where action would clearly be beneficial, it may be wiser to do nothing because the time required to take effective action would be too great in the current network which, for the most part, has no computer support for NM control application.

B.2. RECOMMENDATIONS FOR NM ACTIONS IN NETWORK DAMAGE SITUATIONS

B.2.1. Switch Outages

We define a switch outage as a situation in which a switch is unable to switch calls, nor to signal to or answer neighbor switches, nor can it send reports to a Network Management Center. This situation could result from hardware or software failure in the switch or from physical damage. Our simulation is considerably simplified by assuming that a switch outage event does not interrupt calls already in progress, as they might be in some real outage situations. We believe that this property of the simulated damage makes no significant difference in the effectiveness of the NM actions that are recommended for dealing with the outage situation. It does affect observed network behavior during the transient associated with a damage event, but we assume that in a real damage situation this transient will have subsided before any NM action could be taken, and that it can thus be ignored for all practical purposes.

Switch outage symptoms as observed in our simulated switch reports are:

(1) No report from the damaged switch.

(2) Reporting neighbors show no outgoing attempts from the switch.

(3) If reporting neighbors attempt to use trunks to the damaged switch, the attempts will fail and the neighbors will report reductions in the number of working trunks in any groups to the switch.

The problem of recognizing the outage is made more difficult if there are no reports from some of the neighbors, or if the overall traffic level is low so that an absence of outgoing attempts could occur naturally. In such situations, a decision to declare the switch to be out should be deferred until further evidence from symptoms (2) and (3) appears or other corroborating information, such as a phone call or other message from the site, becomes available. Further, because the recommended control application is a slow and tedious process at present, it is important to ascertain that the outage is likely to persist for many hours or days before carrying out the recommendations.

Our general recommendation for action in a switch outage situation is to apply 100-percent Code Block (CB) controls at all remaining switches for all codes reachable only through the damaged switch, 100-percent SKip (SK) controls on all trunks to the damaged switch from its neighbors, and, in some cases, Reroute controls at appropriate switches.

The purpose of the CBs is to keep traffic that cannot possibly succeed out of the network. Such traffic at Routine precedence level merely wastes some trunk resources, but at higher levels it causes unnecessary preemptions that can significantly worsen the CFR for low-precedence users. With the ruthless, blind-out preemption algorithm employed in DSN, the precedence calls to the damaged node preempt otherwise useful calls but then fail to succeed themselves. The CB application is particularly important when the damaged node is a destination for large numbers of precedence calls. In such a situation, we have seen CFR for routine calls improve from 46 percent to 16 percent by the application of CB. In the same scenario, GOS for routine calls worsened slightly from 66 percent to 69 percent. This reverse correlation, which appears to be counterintuitive, results from the fact that the useless preemptions momentarily open up trunk resources that improve the probability that succeeding call attempts will find free trunks. However, the calls that get these resources are very likely to be preempted by another precedence call attempt to reach the damaged node.

The beneficial effect of the CB application is greatest in a network such as DSN-PAC where the routing provides a rich set of alternatives in most cases. The richness provides more opportunities for useless preemptions and wasted resources. In a net with highly restrictive routing such as DSN-EUR, the application of CB has less benefit since many of the precedence call attempts will fail due to lack of routing possibilities, thereby limiting their opportunities to preempt wastefully.

Our simulations have used a standardized numerical mix of precedences so that the number of precedence calls is proportional to the overall traffic destined for the switch. As a result, we see more benefit from the CB application when we damage a large switch than when we damage a small one. For low traffic to a switch, there is little to be gained from the CB. In the real network we would expect that

some nodes would have a higher percentage of precedence calls than others, so that CB may be indicated even though the overall traffic to the switch is not large. Similarly, when CB is expensive (time consuming) to apply, it is wise to apply CBs first at switches that are likely to be the largest sources of precedence traffic to the damaged switch. With that procedure, the benefits will begin to accrue early in the process, and if the procedure is stopped before completion, the greatest part of the benefit may well have been obtained. The knowledge of traffic patterns needed for such procedures cannot be obtained from switch-report data but must be known *a priori* by the network manager.

The 100-percent SK controls applied to trunks from neighbors to the damaged switch are intended to prevent needless call failures which can occur if the neighbor switches attempt to use such trunks. In our simulations, if the SK controls are not applied and a switch attempts to use a trunk to a damaged neighbor, the call attempt blocks, and the switch reduces the number of working trunks in the group on which the attempt was made. Eventually, the group size will go to zero, and the effect then will be the same as if the 100-percent SK control was in effect. If traffic so routed is high, the capacity may go to zero before the SK controls could be applied, in which case there is no point in making the application. However, when traffic is low relative to trunk group size due to time of day or whatever, it may be a long time before the capacity gets to zero. In such a situation, the SK controls can prevent needless call failures. In the real DSN, if switches do not take trunks out of service on signaling failures or fail to keep them out until NM action is taken to bring them back to normal, then the SK controls are strongly recommended.

We have seen recommendations for the application of 100-percent CANT controls to trunks to a damaged switch as an expedient to accomplish the desired effects of the CB and SK combination. They can be applied much more rapidly than the CBs and have the effect of canceling the calls to the damaged switch as well as preventing the routing of other calls through it. Unfortunately, they have a significant probability of canceling calls that could have succeeded via alternate routes. We have done a number of simulations applying such controls in damage situations and found none in which the CANT controls gave superior performance to taking no control action at all. We therefore recommend against the application of CANT controls in switch-outage situations.

The third component of our recommended prescription for switch outage is the application of Reroute controls or other mechanisms for changing network routing. Our simulations show that routing in the projected Pacific DSN is sufficiently robust that single-switch outages will not cause failure of precedence calls at normal busy-hour traffic levels. However, this robustness is not present in the European DSN due to a routing restriction that in most of the network tandem traffic can take only the primary route out of a switch. The European DSN backbone has a mixture of important hub switches with many neighbors, and other switches with few neighbors. The hub switches appear in the routing topology for many source-destination pairs. An outage of one of these hub switches, or a trunk to such a switch, will cause all calls between certain source-destination pairs to fail independent of precedence level, and service will be very bad for other pairs, again independent of precedence. Since a primary objective of network management is to maintain service for critical users, it is imperative that routing changes be made so that there will be no needless loss of critical communication capability.

Simulations of the projected European DSN have shown that simple routing changes at a few switches can restore full precedence call connectivity in scenarios involving the loss of a single hub switch. Just replacing the first choice route with the second choice is sufficient in most cases to restore successful precedence level communication. In cases where the network topology results in the second choice route merging back into the first before the outage is reached, it may be better to use a third or fourth choice. A more sophisticated routing scheme might be achievable that would improve the handling of routine traffic as well, but we have not yet investigated that possibility.

Simulations of the projected Pacific DSN have shown that, while routing changes are not needed to assure precedence call connectivity in single-switch outage scenarios, two-switch outages do require new routing in many cases.

It should be noted that the above recommendations are appropriate only for an outage that is expected to be of long duration. It makes no sense to begin a control application that might take hours if the switch will be back on the air after a reboot or similar short-duration hiatus. We assume that, in a real network, controllers would confirm (or attempt to confirm) the outage with switch personnel or others local to the switch site before undertaking lengthy control actions. Again, we assume that there would be communication with switch personnel at the time of switch restoral, although such communication is not strictly required since switch reports from the formerly out switch and/or its neighbors will give evidence of its restoral. The CBs and SKs should then be removed, and routing restored to normal.

B.2.2. Trunk Problems

Trunk problems are much more likely to occur than switch problems because there are many more trunks than switches in a network, and transmission facilities are more exposed to environmental changes that can degrade performance. Trunks can fail totally, both as individual trunks and as groups, and they can become noisy, causing users to hang up and retry calls or causing signaling problems between switches. Trunk problems can be of short duration, for example if caused by weather problems in a radio-transmission situation; or they could be of long duration, as might result from the loss of a microwave tower or a satellite terminal. The symptoms may be caused by problems in the transmission medium or in equipment associated either with transmission or with trunk termination at the switches.

Our simulations have shown that, in many cases, trunk problems have little impact on overall network performance and therefore require no NM action other than to make sure that restoral activity is started. However, failure of large trunk groups can result in serious loss of network resources, and our simulations show that, in such cases, trunk outages can have a more severe impact than switch outages because there is no compensating loss of originating traffic.

Since we do not yet simulate noisy trunk problems, we will confine discussion here to trunk outage problems and our recommendations for NM actions to deal with them. Further, we should state that while we have studied only long-duration outages in our simulations, our experience with them suggests that short-duration outages are best left untreated. However, common sense suggests that a sequence of short-duration problems should be treated as a long-duration event of intermittent character, and action should be carried out to treat the problem.

In our simulations, trunk outages can readily be recognized from the switch reports by observing a falling number of reported working trunks and a mismatch between outgoing attempts at one end of a group and incoming attempts at the other. Of course, if traffic is very low, there may be very few attempts, and an outage could go unnoticed for quite a long time. In real networks, controllers often get alarm inputs from transmission equipment (either directly or via phone or message) so that they may be alerted to the problem even before the switch has reported any difficulty. Such inputs may also be helpful in determining whether or not an outage is likely to be of long duration.

If an outage is judged to be of long duration, three NM actions are recommended. First, action should be taken to begin restoration of the lost resources by communicating with the people responsible for the transmission and/or switch maintenance. While waiting for restoration, action should be taken to minimize the damage caused by the loss. For this purpose, 100-percent SKip (SK) controls are effective in preventing fruitless attempts to route calls on problem trunk groups, but their use is recommended only in cases where the outage has been confirmed by inputs from transmission personnel, and there is assurance that the controller will be notified when restoral has been effected. Otherwise, a fractional SKip (say, 90 percent) is recommended so as to allow some attempts at using the trunk group during the outage. When restoral occurs, these attempts will start to succeed and the controller will know that the controls can be removed.

The third recommended action is to deal with the traffic distortion, if significant, produced by the lost resources. In our experiments, we have not found any NM control prescription except routing changes to be effective for dealing with the lost resources. As in the switch outage case, we see differences between the Pacific and European networks. We find no problems with precedence call failures for single-trunk outages in the Pacific DSN with its robust routing, but we observe serious problems in Europe for outages of trunks to the important hub switches. For these cases, routing changes are imperative to achieve precedence call communication for all critical users. The changes must be such as to provide routing paths that do not require the damaged trunks. As in the switch case, experiments show that precedence connectivity can be restored with a few simple changes. However, because a trunk outage reduces capacity without any offsetting loss of traffic, it is likely to produce serious overloading of the remaining trunk resources at heavy-traffic times. The result is often a very poor GOS and CFR for routine users.

B.3. RECOMMENDATIONS FOR NM RESPONSES TO TRAFFIC ANOMALIES

Traffic anomalies can result from normal-traffic patterns offered to a damaged network, or from abnormal traffic offered to an undamaged network. Anomalies are of concern only when they cause resources to become overloaded, with a consequent poor GOS and CFR being observed by some users. Symptoms observed by a network manager in the switch reports are high trunk usage, high percentage of overflows on many trunks, and abnormal percentages of incomplete call attempts. The symptoms may appear over a wide area or only in a small portion of the net.

Three kinds of actions can be effective for dealing with traffic anomalies. The first is to change the network routing to make additional paths available to calls that are failing. The second is to cancel

traffic that has little or no chance of succeeding so as to minimize resources wasted attempting to handle such calls. The third is to use directionalization of trunk resources to relieve the bias associated with a strongly asymmetrical traffic pattern.

Our experiments have demonstrated the effectiveness of routing changes in the European network damage scenarios discussed in the previous sections. Such changes are strongly recommended. In Pacific DSN simulation experiments, we have been able to achieve small performance gains through routing changes, particularly in improving the fairness of service as seen by users at different sites in the network. However, we do not believe the advantages to be gained from such changes are worth the bother and risk involved in the current network control environment.

We recommend cancellation of traffic with Code Block controls when that procedure is indicated. There are two such cases. One is the switch damage case discussed above, where CB at 100 percent is recommended for long-duration outages. The other is in the focused overload situation, where there are a large number of calls to a small set of codes such that most callers are getting busy signals. Our simulations show that a fractional CB can be effective in relieving the symptoms, but we do not recommend such application in the manual NM environment of the current DSN. The overload is likely to have passed before the control application could be carried out.

We do not recommend cancellation of traffic with CANT or CANF controls. Our experiments showed only a few situations in which the controls were beneficial, and those depended upon the notion that giving a canceled call Emergency Announcement (EA) treatment would discourage retry of the call intention. In our simulations, EA treatment does in fact eliminate retries thereby improving GOS in some situations without seriously worsening CFR (the canceled call is treated as a failure). However, in the real world it is not clear that EA treatment will effectively discourage retry. If retries are not discouraged, as they are not in our simulations when No Circuits Available (NCA) treatment is used, we see no beneficial effects whatever from CANT and CANF control actions.

We recommend the use of trunk directionalization via DRZ or DRE controls for the general asymmetrical overload situation. This situation differs from the focused overload discussed above because the incoming calls to the overloaded destination switch are not directed to a small set of codes, but are spread over many lines, so that the ratio of busy to non-busy destinations is more-or-less normal. Such a situation would exist in the case of a natural or man-made disaster when many calls are coming in to inquire about the disaster, making it difficult for users local to the disaster to call out. The symptoms of such situations, as seen in switch reports, are large differences in ICCH and OCCH in combination with more-or-less normal holding times and evidence of call failures at the switch experiencing the higher ICCH values. The evidence of call failures may be directly available in the switch report or may have to be derived from the overflows reported for the trunk groups, depending upon what NM reports are obtained from the switch. Directionalization helps in this situation by making more trunks available for outgoing calls. It should be noted that, with the relatively small trunk group sizes prevalent in the DSN, the DRE control has a very strong effect. Our experiments have shown that reserving just one trunk for incoming calls can reverse quite a large bias. Application of these controls requires continuous monitoring and expeditious alteration or removal when the situation changes.

Fractional Code Blocks are also effective for the general asymmetrical overload, but they are not recommended because of the time required to apply and remove them. In contrast, directionalization is not useful as a treatment for the focused overload situation in which many calls are reaching busy destinations. Our experiments show no significant bias in the CFR experienced by calls into and out of a switch experiencing such an overload if we remove the calls to the busy destinations from the statistics. Directionalization in such a situation introduces a bias where there was none before its application, thereby spoiling performance for users who are not calling the busy destinations.

APPENDIX C
USER'S GUIDES

The Statement of Work provides for delivery of user's guides for the Call-by-Call Simulator and the Network Management Expert System. This documentation has been delivered to DCEC with the latest versions of both CCSIM and NMES, as separate documents to be kept near the machines for user convenience. The purpose of this Appendix is not to reproduce the user's guides themselves, since they are flexible documents that will be changed with each successive delivery of new software versions, but rather to briefly describe the contents of the guides.

The current CCSIM User's Guide consists of six sections, as follows:

(1) Installing the IDSIM Software — tape directory listings and installation instructions

(2) Running IDSIM — how to initialize and run the system

(3) CCSIM Controls — user command vocabulary for controlling a network simulation

(4) CCSIM Network Management Controls — definitions and formats of DSN network management control commands as currently simulated

(5) New Features in CCSIM — documentation of the changes and additions in the current version

(6) Producing Routing Tables for CCSIM — instructions for generating the routing tables needed for the newly implemented Forward Routing and Modified Forward Routing options in CCSIM.

The current NMES User's Guide consists of two sections:

(1) Running IDSIM with NMES — a required supplement to component (2) of the CCSIM User's Guide as noted above

(2) NMES User's Guide — operator interface and command vocabulary.

GLOSSARY

ACH	Attempts per Circuit per Hour — a measure of trunk activity
ACOC	Area Communications Operations Center
ANM	Automated Network Management
ARC	Alternate Route Cancellation
ART	Automatic Reasoning Tool — an expert system shell, product of Inference Corporation
CANF	CANcel-From — a NM control that cancels a fraction of the calls that overflow from a trunk group
CANT	CANcel-To — a NM control that cancels a fraction of the calls offered to a trunk group
CB	Code Block — a NM control that denies entry to the network to calls with specified destination codes
CCS	Common-Channel Signaling
CCSIM	The Lincoln Laboratory Call-by-Call SIMulator
CFR	Call Failure Rate — a measure of network performance calculated as the ratio of failed-to-intended calls
CLLI	Common Language Link Identifier — a naming scheme for interswitch trunks
DAI	Data Acquisition Interface — a component of NMSS
DAMA	Demand Assigned Multiple Access
DCA-Eur	The headquarters of DCA in Europe
DCOSS	Defense Communications Operations Support System
DDN	Defense Data Network
DMCS	DDN Monitoring Center System
DMS	A family of digital telephone switches manufactured by Northern Telecom Inc.
DRE	Directional Reservation Equipment — a NM control that effects directionalization of trunk resources in DMS switches
DRZ	Directionalize — a NM control that effects directionalization of trunk resources in AUTOVON switches

DSN	Defense Switched Network
DTSR	Dial Tone Speed Recording — a measurement of switch performance in providing dial tone to subscriber lines
EA	Emergency Announcement
EISN	Experimental Integrated Switched Network
GOS	Grade of Service — a measure of network performance calculated as the ratio of blocked-to-offered calls
ICCH	Incoming Connections per Circuit per Hour — a measure of trunk activity
IDSIM	The Lincoln Laboratory Interactive DSN SIMulator Katz — a system of computer/analytic tools used by DCEC for network engineering
LLC	Line Load Control — a NM control that denies dial tone to subscriber lines to reduce switch congestion
MAP	Maintenance and Administration Position — a control port on a DMS switch
MC	Monitoring Center
MTFS	Mean Tries for Success — a measure network performance calculates as the average number of attempts needed to complete an intended call
NM	Network Management — in this report refers only to real-time surveillance and network control functions
NMES	Network Management Expert System — a component of IDSIM
NMSS	Network Management Support System — a NM tool developed by GTE for DCA-Europe
NOC	Network Operations Center
NTI	Northern Telecom Inc.
OCCH	Outgoing Connections per Circuit per Hour — a measure of trunk activity
RADR	Receiver Attachment Delay Recording — a measurement of switch performance in responding to interswitch signaling start requests

RR	ReRoute — a NM control that switches parts of routing tables in DMS switches
SK	SKip — a NM control that causes a fraction of calls offered to a trunk group to skip over that group in the routing chain
SM-ALC	Sacramento Air Logistics Center
TAC	A computer that provides user access to a data network of the ARPANET type, but does not itself provide computational resources
TCP/IP	The official DoD protocol suite for internet data communications

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) K-BSACDSNTA	5. MONITORING ORGANIZATION REPORT NUMBER(S) ESD-TR-88-314
6a. NAME OF PERFORMING ORGANIZATION Lincoln Laboratory, MIT 6b. OFFICE SYMBOL *(If applicable)*	7a. NAME OF MONITORING ORGANIZATION Electronic Systems Division
6c. ADDRESS *(City, State, and Zip Code)* P.O. Box 73 Lexington, MA 02173-0073	7b. ADDRESS *(City, State, and Zip Code)* Hanscom AFB, MA 01731
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Air Force Systems Command, USAF 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-85-C-0002
8c. ADDRESS *(City, State, and Zip Code)* Andrews AFB Washington, DC 20334	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62702F; PROJECT NO. Program 295; TASK NO.; WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*

Knowledge-Based System Analysis and Control Defense Switched Network Task Areas

12. PERSONAL AUTHOR(S)
Harold M. Heggestad

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Annual Report	FROM 1 Oct 87 TO 30 Sep 88	1988, September, 30	82

16. SUPPLEMENTARY NOTATION

None

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUB-GROUP	technical control; network management; expert systems; knowledge engineering; machine intelligence

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

An Interactive Defense Switched Network Simulator (IDSIM) has been implemented, consisting of an enhanced Call-by-Call Simulator (CCSIM) in one computer interfaced with a Network Management Expert System (NMES) in a second computer. The operation of IDSIM is similar to that of a real-world theater Defense Switched Network (DSN) and its community of users, at a future time when the DSN is fully installed and an Expert System at each theater operations center performs DSN network management (NM) functions. Within IDSIM, NMES collects periodic activity reports from each simulated DSN switch in CCSIM, analyzes them to identify network problems, and applies control commands to the simulated switches to circumvent the problems as well as possible. By inducing network fault and overload conditions, applying controls, and studying the results, an experimenter can exploit IDSIM as a system engineering tool to develop NM strategies for the DSN. This report describes IDSIM as well as a set of NM study results obtained with this tool. The report also describes the applicability of CCSIM as a near-term training device for human DSN NM operators, and analyzes expert system applicability to Defense Data Network (DDN) management.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Lt. Col. Hugh L. Southall, USAF	22b. TELEPHONE *(Include Area Code)* (617) 981-2330	22c. OFFICE SYMBOL ESD/TML